DFP: AUTOMATED MEASUREMENT SYSTEM OF PCB

11/05/2018

Project Report

Degree in Industrial Electronics and Automation

Supervisor: Josep Font Teixidó

Student: Sergio Lorente Espigares

1 Contents

LIST OF FIGURES

LIST OF TABLES

1 ABSTRACT

The automated measurement system of pcb project (hereinafter, AMSOP), is intended to develop an automated system which improve the way to carry out different tests in a printed circuit board (pcb). The project mainly focuses on developing a LabVIEW VI (LabVIEW application) that allows to control a physical coordinate system (System) through hardware components to measure electric voltage of a pcb. The data will be collected and stored to use it in future reports. More specifically, the AMSOP is created for the users who must take repeated measures and/or during a long time.

The project is carried out by Mr. Sergio Lorente Espigares in the framework of Ficosa International S.A., with the help of Mr. Josep Font Teixidó. With different backgrounds and technical skills, I am going to work during the following semester on AMSOP, that will help the user to take measures efficiently, saving time and resources by removing repetitive work. In the end, it is expected that the AMSOP will be an instrument able to guarantee of productive data collection.

2 INTRODUCTION

2.1 PERSONAL MOTIVATION

The achievement of this project supposes to me, at a personal level, the opportunity to develop a new system that attends a need.

The opportunity to go into a project of this characteristics will test me and will make me translate all the knowledge learned in college and will allow me to be involved in new ones.

I also believe that I will learn to manage a project which touches on such essential aspects of a technological product and to learn from the different tools used to realize it.

2.2 AIMS

Due to the dimension of the project, the aims are divided into two parts. The following project has both hardware and software objectives. On the one hand, the software objectives, which are listed below:

Creation of a LabVIEW application (Virtual Instrument (VI)):

- ➔ Stablish communication between Arduino and the program.
- ➔ Custom front panel for the user interface (UI).
- ➔ Control and information about the System.
- ➔ Real-Time information of the measures.
- ➔ Export the data to facilitate its subsequent use.

On the other hand, it is possible to define some hardware aims including the building of the System (mechanical) and the electronics.

System building (Mechanical):

- ➔ Initial schematics and blueprints.
- ➔ Physical system construction.

Electronics:

- ➔ Components and electronic devices research.
- ➔ Schematics of the project.
- ➔ Mounting and disposition of the different components and support devices.

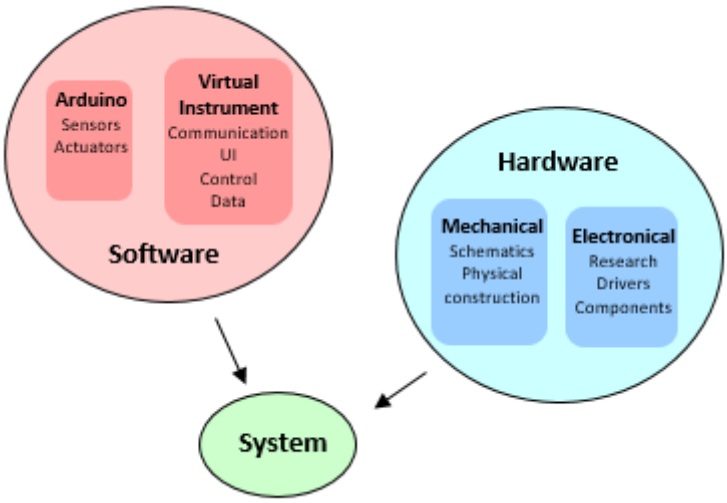


Figure 1: HARDWARE AND SOFTWARE FOR THE PROJECT

2.3 TIMETABLING

To achieve the success in the end of the semester it is important to plan the work process ahead. Therefore, a Gantt diagram was created, which gives an idea of how has been working during this time.

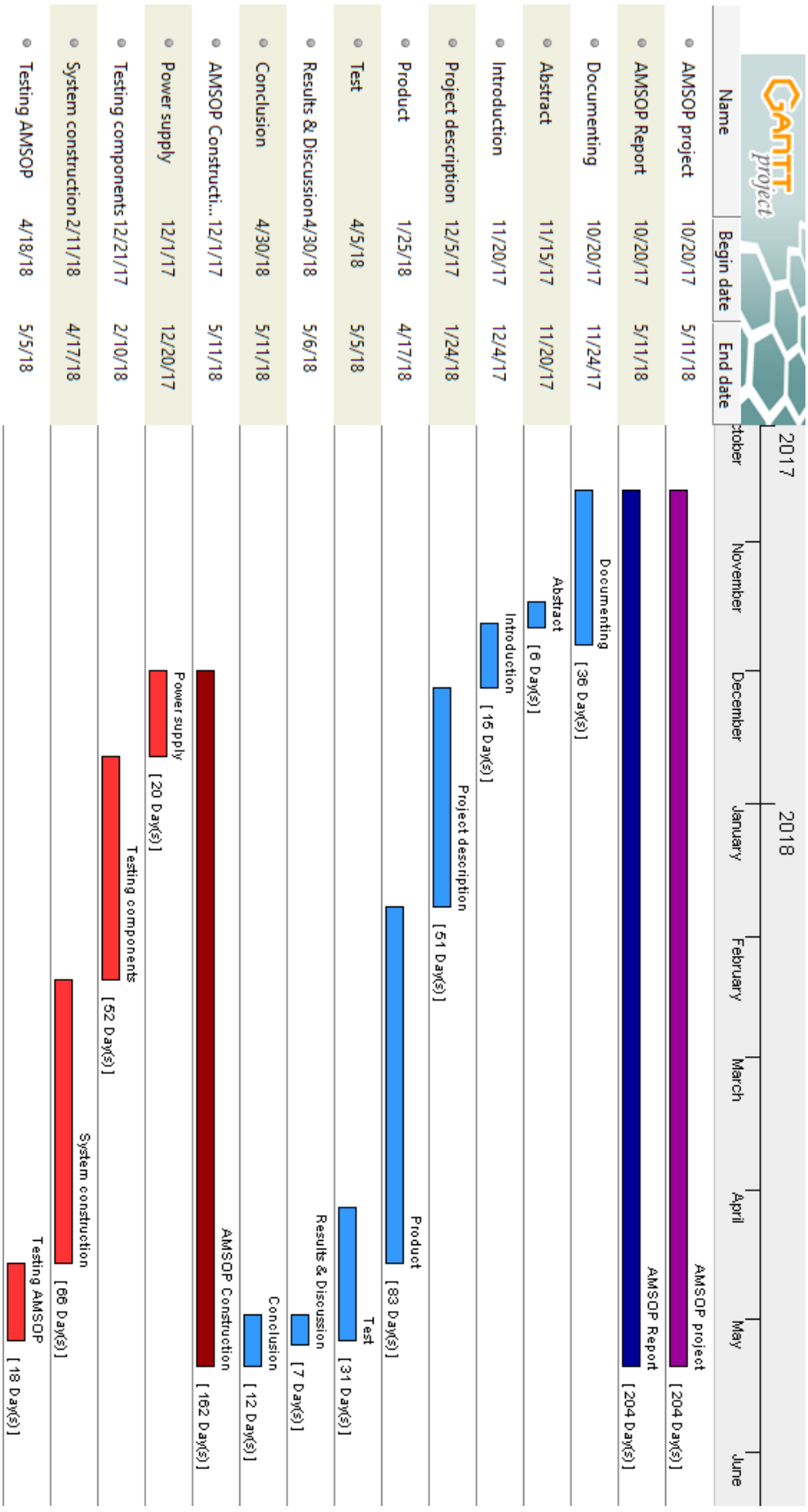


Figure 2: AMSOP GANTT DIAGRAM

2.4 DEVELOPMENT TOOLS

In this section there are explained the development tools used in this project, and for what had been used them.

Microsoft Word

Microsoft Word is a word processor which allow user to write text. The program was used to write all the report.

NI LabVIEW 2017

NI LabVIEW 2017 enables to create a LabVIEW applications (VI). Was the program used to build the application.

Arduino Software (IDE)

Arduino Software (IDE) enables to make a code and debug. Was used to charge the pre-configured program to Arduino.

Google Drive

This is a cloud storage that enables to upload files to save it. Was used to save files in the cloud.

Draw.io

This is online diagram software for making flowcharts, process diagrams and network diagrams. Was used to draw some graphic things like BDD, IBD and state machine diagram.

EasyEDA

This is free, online and easy to use circuit design. Used to make the electronic schematics.

GanttProject

This program allows to make easily a Gantt diagram. Used to make a Gantt diagram of the project.

3 PRELIMINARY ANALYSIS

In this chapter, it is discussed about the choices taken for the project. First, it had to choose the motors that are going to move the System (X, Y and Z axis), the drivers used to restrict the current through the motors and the endstop switches that will help to find "home" position. It also had to choose a microcontroller which would fit with the expectations. Then, it is discussed about what is the best way to power all the System, how to construct the mechanical part and finally, how to measure the voltages and why it has been chosen LabVIEW as an application builder.

3.1 HARDWARE AND SOFTWARE DISCUSSION

3.1.1 Electronical

3.1.1.1 Motors

Attributes	**DC Motor**	**Servo Motor**	**Stepper Motor**
Control	PWM	PWM controlled	Steps
Accuracy	Low	High	High
Rotation angle	360º	180º	360º
Rotation	Continuous	Continuous/Controlled	Controlled
Price	40€	12€	17€

Table 1: COMPARAISON TABLE OF MOTORS

For the project, is needed to use motors to move the axis of the System. Actually, it is needed three motors to move X, Y and Z axis. As can be seen in the table above, due to the principal necessity of control and accuracy of the movement, the DC motor was discarded. Apart from that, the System needs a 360 degrees rotation angles from the motors so that by converting that rotational movement into a linear one, it can reach as much distance as possible. The main required characteristic is that is needed a motor which can be controlled all the time and therefore DC and servo motors have been discarded. Finally, it has been chosen the stepper motor, that in addition to offering everything mentioned, has a good torque that will make possible to move the System effortless.

Once chosen the kind of motor (stepper motor), it is having to choose between the great quantity of motors that the market offers. Due to that the project is like others as CNC or 3D printers, it is selected a standard NEMA.

To select the motor, the following requirements have been considered:

- Torque (strength and speed)
- Adequate current consumption

Because of the load they are going to move is considered "average", the minimum value to use is 40N·cm. In the following section, the choice of the driver is assessed but we can say that for an optimal control of the motor, the nominal consumption has not to be very high. It is advised that the nominal current should be 1.7A or 1.8A per phase. With these values, the best option is the motor Nema17.

Figure 3: STEPPER MOTOR NEMA 17

3.1.1.2 Drivers

The current limiter of the motors is the driver. The importance is given because it simplifies the control of the stepper motor from an automata or processor like Arduino. Those controllers allow to handle high voltages and current that motors need, the drivers restrict the current that circulate through them and provide the necessary protection to avoid electronic damage.

To control them, basically it is required two digital outputs of the microcontroller, one of them is used to indicate the direction of the rotation and the other one is used to communicate that is wanted to take a step. Also allow the microstepping to get superior precision of the motor nominal step.

A typical driver comparation is A4988 and DRV8825, being the last one an improved version as can be seen in the following table:

Name	**A4988**	**DRV8825**
Fabricant	Allegro MicroSystems, LLC	Texas Instruments Inc.
Operational Voltage	8 – 35 V	8,2 – 45 V
Microstepping allowed	Full-step, 1/2, 1/4, 1/8, 1/16	Full-step, 1/2, 1/4, 1/8, 1/16, 1/32
Maximum coil output current	2 A peak; 1,4 A RMS	2,5 A peak; 1,75 A RMS
Overcurrent protection	Stop > 2,1 A per coil	Stop > 3 A per coil
Short-circuit protection	YES	YES

Table 2:COMPARAISON TABLE OF DRIVERS

The principal differences between drivers are that DRV8825 can work with high voltages and currents (45V in front of 35V and 2.5A in front of 2A). Besides that, add another microstepping mode (1/32) which is not included in the A4988 version.

As can be noted from above chart, the components have overcurrent protection as well as short-circuit protection.

These drivers are really economics, an A4988 may cost over 6€ the five units and the DRV8825 around 10€ the five units. So, the fact that the DRV8825 is a new version of the A4988 with improved characteristics and the insignificant difference of price make DRV8825 the best choice to use in this project.

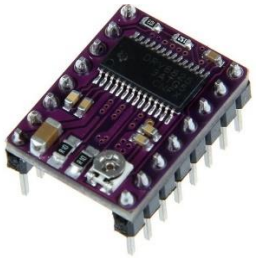


Figure 4: DRIVER DRV8825

3.1.1.3 Endstop switches

The main problem using the NEMA17 motors is that there is no way to know in the position they are once they start to run. This supposes a big problem in the System, where is needed a coordinate system. As a rule, in this kind of projects is necessary to know the position (0, 0, 0) from the start of the program to be correctly located during its operation. This is achieved through the endstop switch, which indicate the System where is the starting position or “home” and, consequently, is going to calculate the maximum number of steps allowed in the three axes. There are several endstop switch models from the simplest, the mechanical endstop switches, to the most complex and sophisticated ones like optics or even inductive or capacitive, among others.

On the one hand, as an advantage and because it offers an easy way of installation, robustness of the system, operation on high voltage and immunity from static electricity is going to work with the mechanical endstop switches.

On the other hand, the disadvantages are the detection speed and the possibility to having contact bounces, furthermore, it depends of the actuation force.

Figure 5: MECHANICAL ENDSTOP SWITCHES

3.1.1.4 Microcontroller

Attributes	**Raspberry Pi**	**Arduino UNO**	**Arduino MEGA**
Model	Model B	UNO	Mega
Microcontroller	64-bit Quadcore Cortex A7; GPU Dualcore videocore IV	8-bit ATMega328	8-bit ATMega2560
Input Voltage	5V, 2A	Recommended 7 – 12V (limit -> 6-20V)	Recommended 7 – 12V (limit -> 6-20V)
I/O	40	20 (6 for PWM and 6 for Analog)	70 (14 for PWM and 16 for Analog)
Clock Speed	CPU 900Mhz	16Mhz	16Mhz
USB	X4 (2.0)	X1	X1
Programming language	JAVA, PYTHON, C… This is a microcomputer with a Linux distro	Arduino's own language (based on C and Processing)	Arduino's own language (based on C and Processing)
External power supply	YES	YES	YES
Memory	1Gb	32Kb	256Kb

Table 3:COMPARAISON TABLE OF MICROCONTROLLERS

In this project the microcontroller is the most important device because is the responsible to send and receive signals, activate and deactivate inputs and outputs, in essence, is the bridge to control all the System. To control all the components of the System is necessary the inclusion of a microcontroller. In this case, the main idea is to use one of the table above.

Due to Raspberry pi is more a computer than a simple controller the idea is to use an Arduino, Uno or Mega. The difference between both possible boards lies with the capacity of memory and the number of analogic and digital inputs and outputs. Arduinos platform comes with his own software and it is quite simple to use it.

Thinking in the project, where is not necessary to monitor many signals the first option is Arduino Uno that it is the most economical and perfectly fulfils with the necessities, but in this case, is going to use Arduino Mega, which is already bought and can give the flexibility of, in the future, could expand the project adding some improvements.

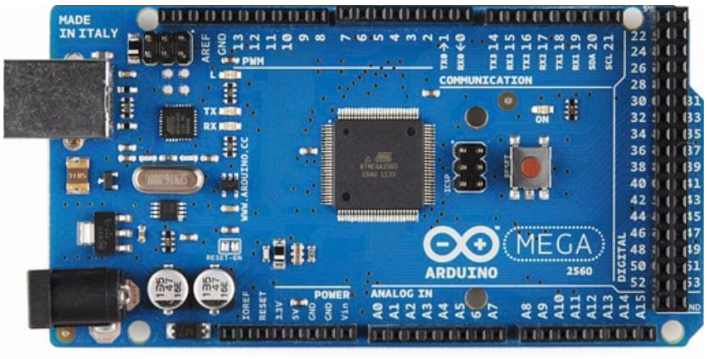


Figure 6:ARDUINO MEGA

3.1.1.5 Power source

This section is intended to evaluate the best way to power all the System. As it has been seen above [*section 3.1.1.1 Motors*], to supply all the circuitry is required a minimum of 12V and around 15A and unfortunately it is impossible to obtain this values without an external power supply. Several options were considered, to buy a laboratory power supply or modify one of a PC ATX working as a laboratory one. Obviously, the best choice is directly to buy a new one but was considered an interesting and economical idea to modify an old ATX PC power supply. In the following sections is explained the procedure to do it.

Figure 7:ATX POWER SUPPLY

3.1.1.6 DAQ

When is desired to measure voltages in the board, to test it, is necessary any device that can take the samples in the real world (analogic system) and generate the data that can be manipulated by a computer or other electronical device (digital system). Basically, consist taking some physical signals or electrical voltages and digitalize them in a way that can be processed by a computer. The element which do that is data acquisition board (DAQ). It is not going to talk about how it works or the way the board process the data.

In the market there are a lot of DAQ devices and with different characteristics that make them perfect to use it in a specific case. These devices are quite expensive but fortunately it is possible to use one of them for the porpoise of the project. Specifically, the device is as follows:

Name	**NI PCIe-6341**
ANALOG INPUT	
Number of channels	16
Simultaneous	NO
Sample rate	500 KS/s
Resolution	16 Bits
Absolute accuracy	2.19 MV
ANALOG OUTPUT	
Number of channels	2
Update rate	900 KS/s
DIO	
Number of channels	24

Table 4:DAQ NI PCIe-6341 CHARACTERISTICS

The main differences between the DAQ boards are the input/output analogic/digital channels, the resolution, sample rating connectivity and counters. There are a lot of DAQ diversity so it is possible to choose the best one for the desired purpose.

The DAQ of National Instruments is perfectly imbedded with LabVIEW, that makes it easy the acquisition and manipulation of data. It is going to use this device to take voltages and streamline the subsequent signal processing process.

Figure 8:DAQ NI PCIe-6341

3.1.2 Mechanical

3.1.2.1 Structure

As regards mechanical part of the project, in the first was initially considered to create on an autonomous way all the structure of the System but that would be an enormous quantity of time adding the difficulty not having the right tools to take the cuts in the pieces. Finally, was ordered the design to a specialised company. The structure is like a 3D printer or CNC machine.

The structure is one of the most important part of the project cause if the structure is not stable it is possible to fail when the samples are measured as a minimum movement of the structure would affect the X, Y and Z axis.

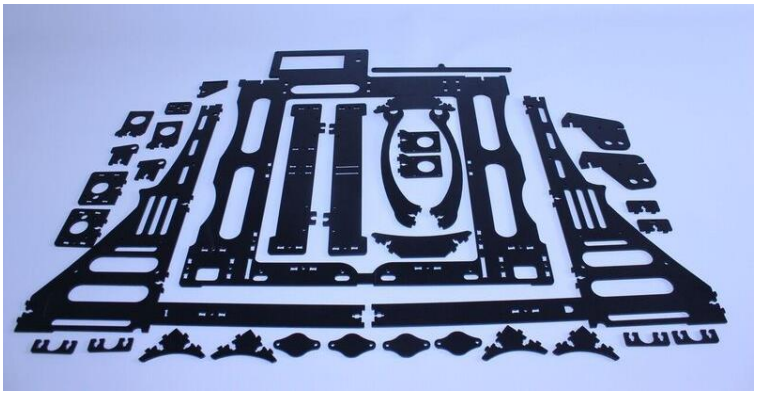


Figure 9:PARTS OF THE STRUCTURE

3.1.3 Components

The components for the mechanical part has been obtained from specialized web of 3D printers. It was not difficult to collect all the materials because the System is really similar to a 3D printer or a CNC machine. The list of mechanical components is the following:

- Straps
- Pulleys
- Rods
- Bearings
- Nuts
- Washers
- Screws
- Couplers

Figure 10:COMPONENTS OF THE STRUCTURE

It is necessary to have greased bearings and straps always tight so that there is no excess friction or jerks when using the machine.

3.1.4 Software

3.1.4.1 Arduino

As has been chosen Arduino as the main microcontroller it will be reasonable to use Arduino IDE software. This software is based on C language and is easy to write programs and upload them to the board. It runs on different operative systems i.e. Windows, Linux and Mac OS X which allows use it without complications. Arduino IDE supports many different libraries and has basic coding examples. For the purpose of reading and write the data it is believed that this software is the best one to use.

However, the idea is to control Arduino through LabVIEW but this is would not be possible without LINX, an open source project of Digilent, designed to facilitate the development of the embedded applications using LabVIEW. LINX allows access to a peripherals as digital I/O, analogic I/O, PWM, I2C, SPI and UART which is perfect for the project because it is intended to use Arduino Mega as a microcontroller of the sensors and actuators.

LabVIEW LINX is going to control Arduino through USB/Serial connection. To use it is needed to install the firmware in Arduino, it is possible to know how to do it in the web page of National Instruments. With the firmware installed, then can be directly work in real time on Arduino, visualize the data which is working on and create an application quickly.

3.1.4.2 LabVIEW application

LabVIEW is a programming environment, which is used to create programs using a graphical notation (.G language). It differs from traditional programming languages like C, C++, Java… where is programmed with text. LabVIEW offers more flexibility than standard laboratory instruments because is software-based. It is possible to create an exactly type of instrument needed due to the virtual instruments (VIs).

There are many libraries for functions and subroutines to help in different tasks with different purposes. Is possible to make the connection with data acquisition boards (DAQ), general purpose interface bus (GPIB), serial instrument control and communication over internet. That is why LabVIEW is the main tool used in this project.

The main part of the project is to create a LabVIEW application where user could interact directly with the System.

The application will manage, on the one hand, Arduino and the movement of the motors and reception of the endstop switches responses and, on the other, will manage the data coming from the DAQ (voltages of the board), as well as data manipulation to generate the final reports. All these functions must be properly programmed because they may interfere with the procedure each other.

4 REQUIREMENT SPECIFICATIONS

4.1 OPENING

4.1.1 Purpose

Shall be given a general description about the project through this section. The major objective of this project is to enable user to check the test points of the pcb through a LabVIEW application (.vi). The users can receive information about the status of the program through application. It is going to make a connection between microcontroller and the application for a mutual information transfer. The microcontroller will send signals measured from the electronical components (endstop switch) to the application and the application will send to the microcontroller signals to move the actuators (motors) and to know where they are. Finally, users can easily know how it is going the process, voltage averages and control it.

4.1.2 Reading Instructions

4.1.2.1 General description

General project description.

4.1.2.2 Use cases

Descriptions about use cases of the system.

4.1.2.3 Non-functional requirements

Descriptions about the non-functional requirements of the system.

4.1.2.4 External Interface requirements

Descriptions about the user interface.

4.2 GENERAL DESCRIPTION

4.2.1 AMSOP description

4.2.1.1 AMSOP overview

AMSOP project is a system created to test the boards for a long period of time. The project consists of electronic components that make possible the control of the motors in the System and a LabVIEW application that displays the data captured by the DAQ and make a report of the measurements. As a feedback loop, the application has the functionality to show the process and sends alerts if the process has suffered a failure, so that user can support the System. It is also important to use microcontroller which will make the communication between the system and the application.

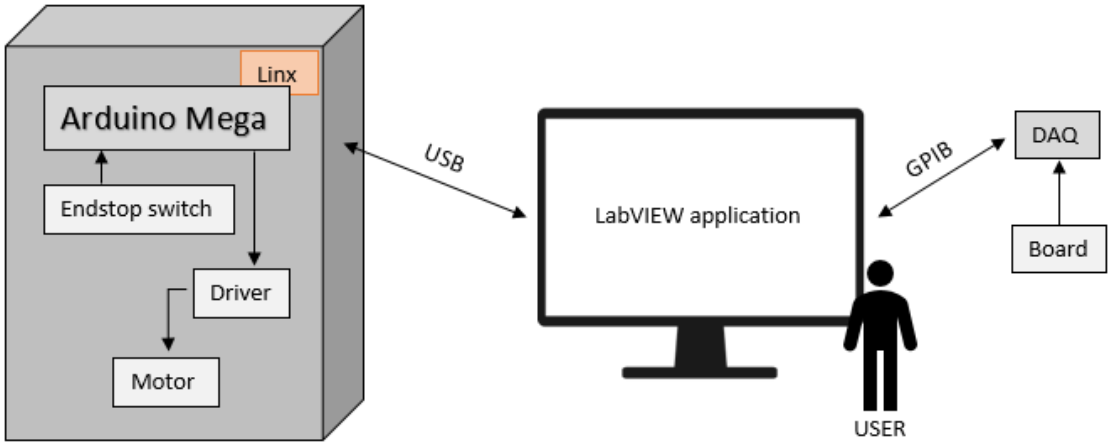


Figure 11: GENERAL AMSOP OVERVIEW

For the System it is used one type of sensor which measure when the motor has arrived at the final position (endstop switch). The sensors are connected to Arduino Mega board with which it is possible to send the information of the sensor to the application. Furthermore, it is used drivers to manage the motors as wanted. To make the System communicate with the application it was decided to use Serial/USB communication (the most common). However, the DAQ communication is via GPIB (integrated in the computer). Both kind of communication require that the computer must be close to the system.

The application itself represents the user-friendly interface where the voltages and the status of the program will be displayed. The data is shown in real-time so that the person can see the measurements at any time in case something is not as it is required and act accordingly.

The crucial part of the user interface is to give the feedback about the process. For example, showing the current process state of the program or being advised if any failure occurs. If the coordinate format is not correct, a message is sent to correct it. If a long period of time has passed and there is still a lack of data to analyse, the user is notified.

4.2.1.2 Actor-Context diagram

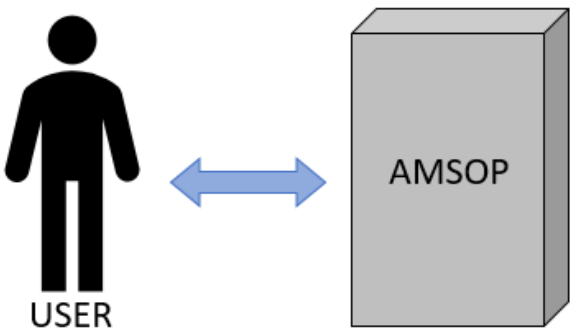


Figure 12:ACTOR-CONTEXT DIAGRAM

4.2.1.3 Use Case description

User's role: Is possible to define the user as the primary user that, using the information that arrives from the AMSOP, is able to take decisions and make actions using the LabVIEW application.

AMSOP: Place provided of electronic components that makes possible for the User to control the System and at the same time, is able to notify the User different circumstances.

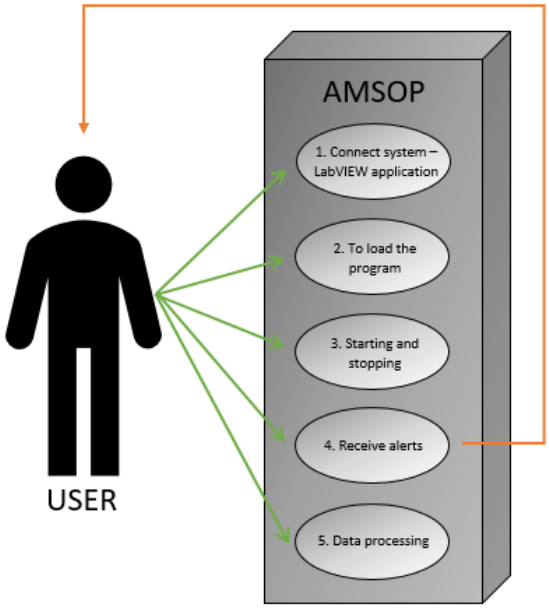


Figure 13:USE CASES DIAGRAM

4.3 USE CASES

4.3.1 Use case 1

Name	Connect system – LabVIEW application
Use case - ID	UC 1
Aims	Get a connection between the system and the application
Pre-conditions	The pc and the system are running
Post-conditions	The system and the application are connected
Main scenario	1. If the system (Arduino) and LabVIEW application are not connected, the user must verify the USB connection and connect the cable correctly. 2. The system and the application are correctly connected. The user must open the application (.vi) and click run button ⇨ .
Exceptions	1. The system is not powered. 2. The PC is not powered.

Table 5:UC 1

4.3.2 Use case 2

Name	To load the program
Use case - ID	UC 2
Aims	To load the coordinate program in the application
Pre-conditions	The application is running
Post-conditions	The coordinate program is loaded
Main scenario	1. The application will show a selection window. 2. The user selects the right program (.txt format).
Exceptions	1. Wrong format chosen.

Table 6:UC 2

4.3.3 Use case 3

Name	Starting and stopping
Use case - ID	UC 3
Aims	Start and stop the AMSOP
Pre-conditions	The application is running and the system is correctly powered.
Post-conditions	The AMSOP is correctly started and stopped.
Main scenario	1. The user clicks START/STOP button. 2. The user visualizes how the system response is generated.
Exceptions	1. Wrong connection between the system and the application. 2. Invalid reading of coordinates.

Table 7:UC 3

4.3.4 Use case 4

Name	Receive alerts
Use case - ID	UC 4
Aims	Get the information of the issue in the screen
Pre-conditions	The application is running
Post-conditions	The alerts are displayed on the screen
Main scenario	1. An alert is shown in the screen. 2. The user must stop the program and try to fix the issue before to continue.
Exceptions	1. The application or the system are correctly working. 2. The alert is not correctly registered.

Table 8:UC 4

4.3.5 Use case 5

Name	Data processing
Use case - ID	UC 5
Aims	The data is being processed
Pre-conditions	The application is running and the system is correctly powered.
Post-conditions	The data is correctly processed
Main scenario	1. The user can move through the application tabs. 2. The voltages and coordinates are displayed in their different tabs in real time.
Exceptions	1. Wrong connection between the system and the application. 2. Incorrect reading of data.

Table 9:UC 5

4.4 NON-FUNCTIONAL REQUIREMENTS

4.4.1 Motion

For the project is it is necessary to control the motors to move the system. The idea of automating the system makes the motion speed a non-functional requirement where 2 seconds of delay is tolerated since the order has been given until the motor begins to move.

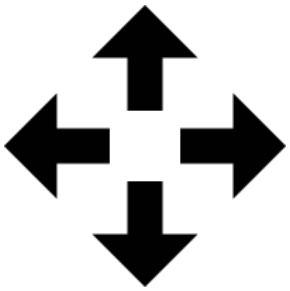


Figure 14:MOTION

4.4.2 Voltage

About voltage, the only useful criteria is the accuracy, which cannot exceed an error of ±0.005 Volts. Any other value would be not accepted in the final report.

Figure 15:VOLTAGE

4.4.3 Temperature

The temperature reached by the driver (motor limiter) must be considered because it is known that if the temperature exceeds 165ºC the driver is going to turn off. Maybe it is interesting to add a fan in the system if the temperature is close to be reached.

Figure 16:TEMPERATURE

4.5 USER INTERFACE REQUIREMENTS

4.5.1 Overall LabVIEW application description

The following chapter will describe the LabVIEW application which aims to displays the voltage measurements taken of a pcb specific test points and allows the user to control the sampling time and sampling rate.

The figures below illustrate the overall design of the LabVIEW application with 2 tabs, “Main Page” and “Measurements”.

When the user just opens the LabVIEW application, a message appears indicating the procedure to continue (Figure 17). The user has to select the corresponding program (Figure 18) and click “OK” button in the front panel. Once the user has pressed “OK” button, the program is going to load and the image of the corresponding pcb is going to update as well (Figure 19).

To start the program, is needed to fill the name of the sample because will be the name with which the report will be generated and click “START” button.

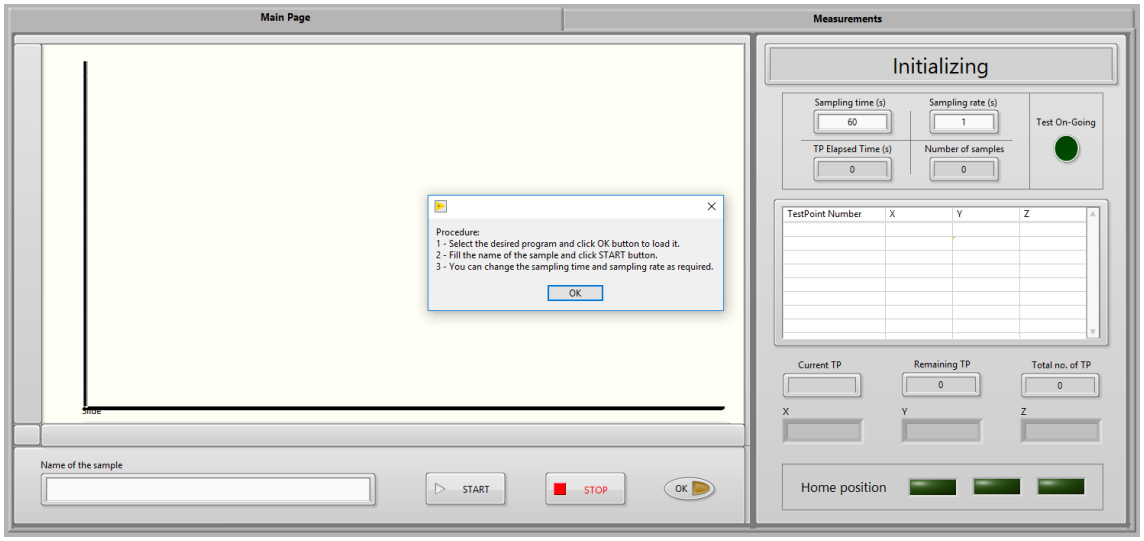


Figure 17:FRONT PANEL MESSAGE

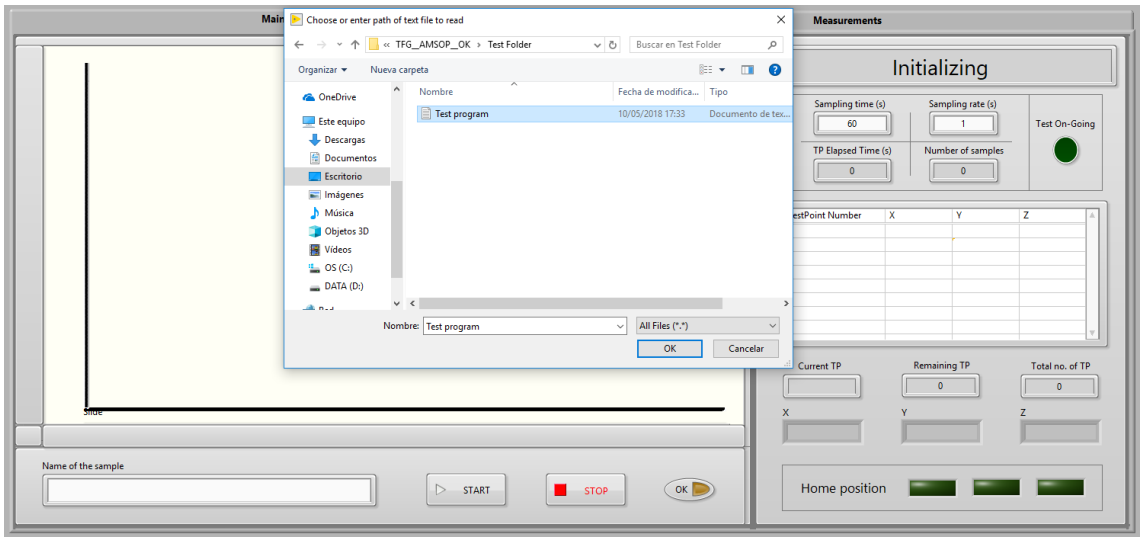


Figure 18:FRONT PANEL LOADING THE COORDINATE PROGRAM

The “Main Page” tab (Figure 19) displays the uploaded values of the coordinate program and indicates the current test point checked and its coordinates, how many test points remains to check and the total number of test points loaded in the program. The user can change as required the sampling time of the program.

The program automatically will test the different test points showing its state in the status indicator. The movement of the motors will be represented in the plot as well.

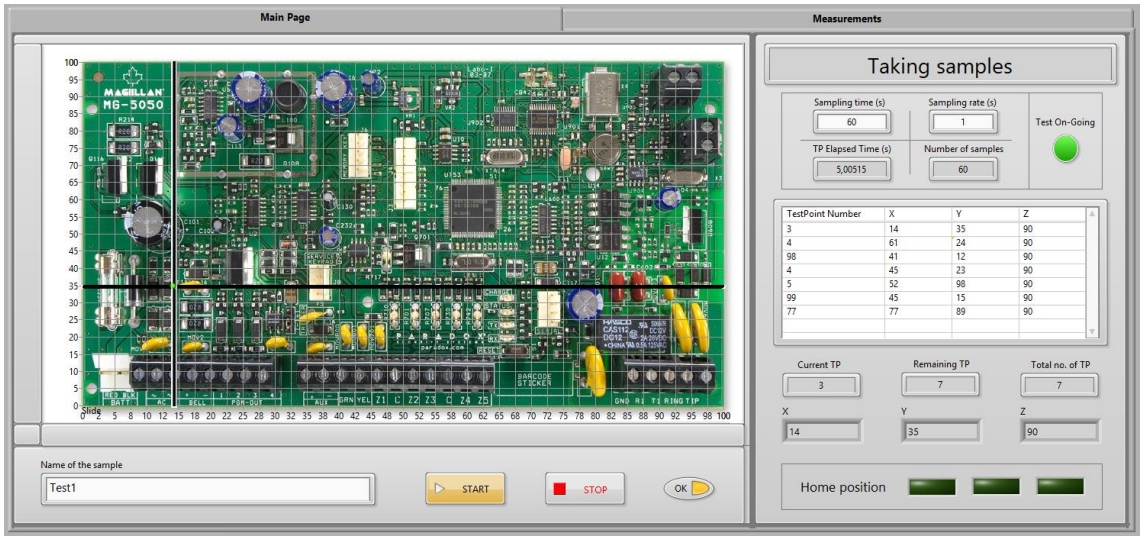


Figure 19:MAIN PAGE TAB

Furthermore, there is the “Measurements tab” (Figure 20) which it gives a resume about the measurements taken and it displays the current voltage of the test point.

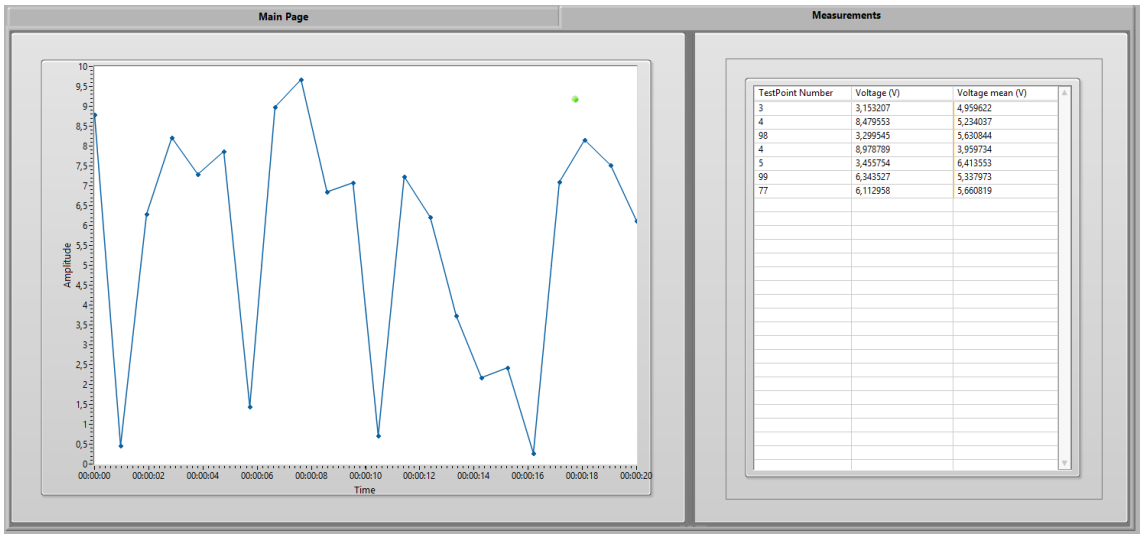


Figure 20:MEASUREMENTS TAB

The user can stop the program whenever required clicking on “STOP” button.

Once the program has been stopped, a report with the name filled in the “Name of the sample” will be generated (Figure 21). The report has six columns:

- **Time:** shows the time when the sample was taken.
- **No. of TP:** Shows the number of the test point.
- **Voltage (V):** Shows the voltage of the sample.
- **Voltage mean (V):** Shows the mean of the voltage. Every row represents the next test point.
- **Max. Voltage (V):** Shows the maximum value of the voltage registered. Every row represents the next test point.
- **Min. Voltage (V):** Shows the minimum value of the voltage registered. Every row represents the next test point.

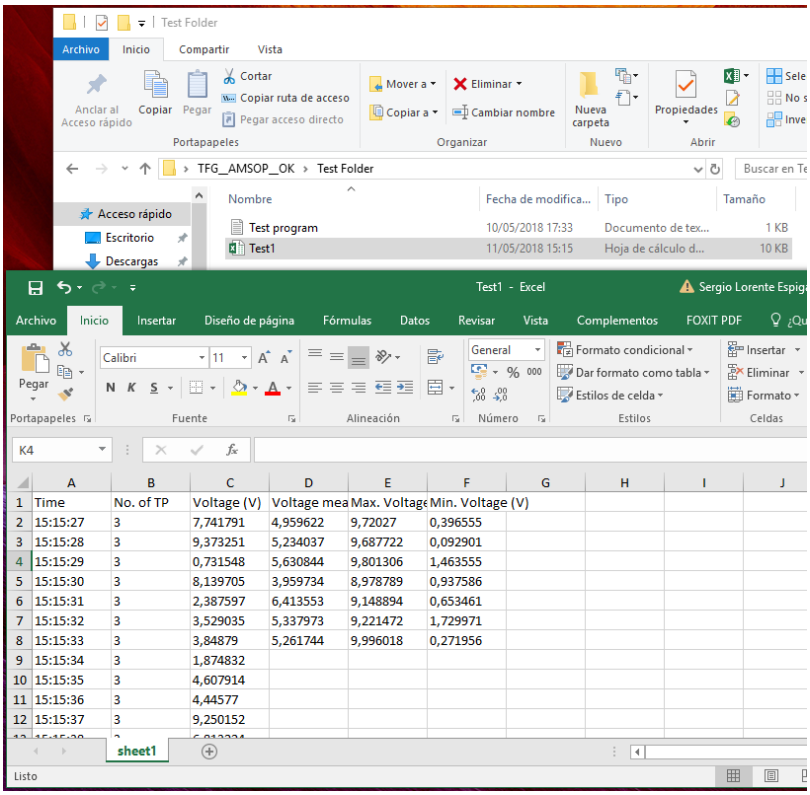


Figure 21:REPORT GENERATED

In addition, an image of the pcb used with the test points checked is created as well (figure 22).

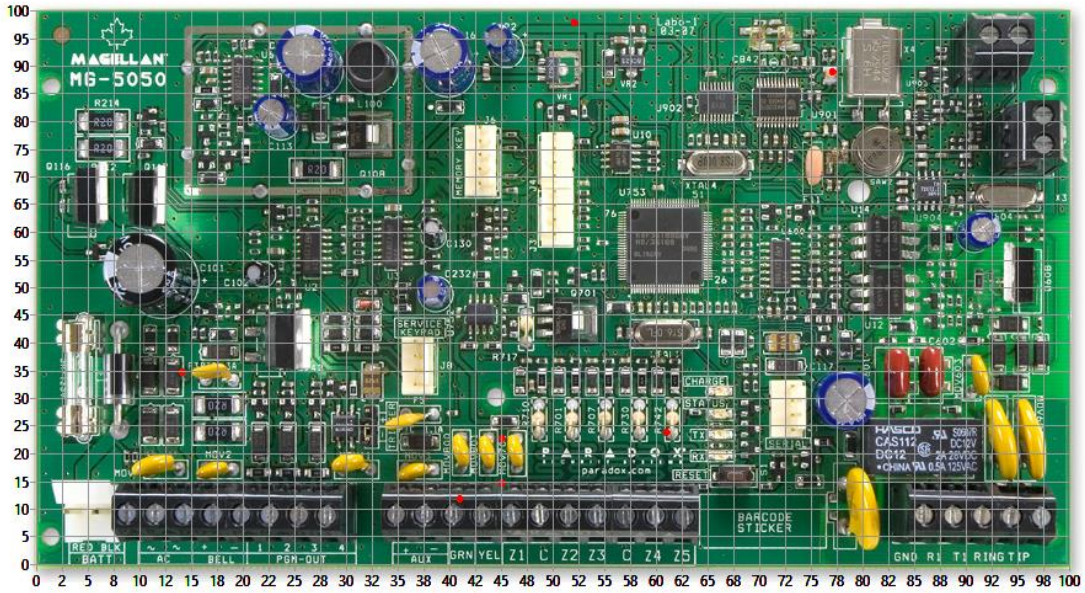


Figure 22: PRINTED IMAGE OF PCB WITH THE TEST POINTS

5 HARDWARE ARCHITECTURE AND DESIGN

This section describes the hardware architecture of the AMSOP system. The following chapters will show the main components: The Arduino microcontroller, the electronic components that it is going to use, the drivers and the power source to supply all the System.

5.1 HARDWARE ARCHITECTURE

This section's aim is to explain how the hardware architecture is going to be. The BDD (Block Definition Diagram) and the IBD (Interaction Block Diagram) will be used so that to show how the system is going to work.

5.1.1 AMSOP

5.1.1.1 BDD (Block Definition Diagram)

The figure below describes the AMSOP using a BDD diagram.

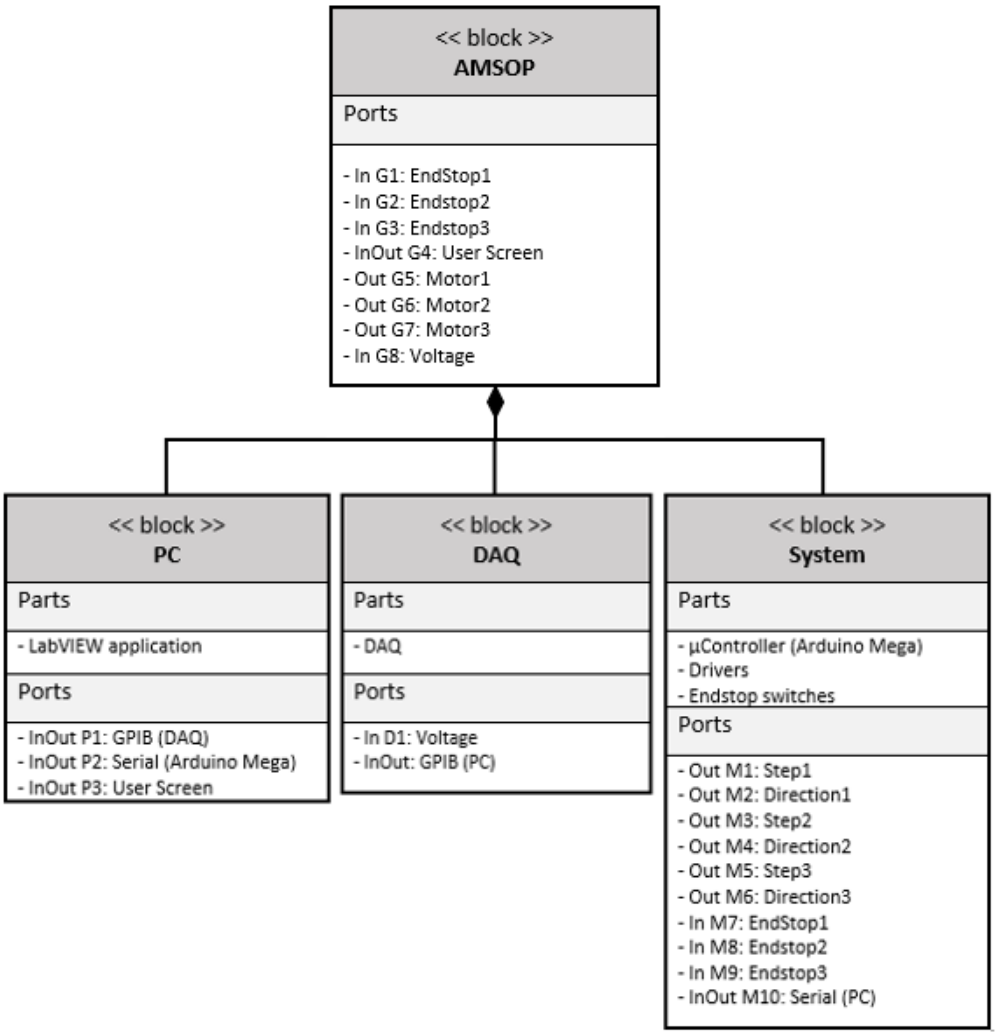


Figure 23:BDD OF THE AMSOP

BLOCK DESCRIPTION

In this table, all the block names and functions are described; the ports associated to each function are described and commented, and their directions are indicated.

Block Name	Function Description	Port Name	Direction	Comment
AMSOP	This block shows the combination of the devices that compose the block.	G1 (EndStop1)	In	Port to receive the information of the Endstop switch 1
		G2 (EndStop2)	In	Port to receive the information of the Endstop switch 2
		G3 (EndStop3)	In	Port to receive the information of the Endstop switch 3
		G4 (User Screen)	In/Out	The user controls the system through the front panel (LabVIEW application screen)
		G5 (Motor1)	In	The motor will be managed by the driver 1
		G6 (Motor2)	In	The motor will be managed by the driver 2
		G7 (Motor3)	In	The motor will be managed by the driver 3
		G8 (Voltage)	In	Port to receive the voltage data taken by the DAQ
PC	This PC has the LabVIEW application to control the system.	P1 (GPIB)	In/Out	Using GPIB connection is stablished the communication of PC - DAQ
		P2 (Serial)	In/Out	The user controls the Arduino through LabVIEW application with serial communication
		P3 (User Screen)	In/Out	The system is going to communicate with the user through the front panel (LabVIEW application screen)
		D1 (Voltage)	In	Port that is going to receive the voltage

DAQ	DAQ is the part that has the DAQ (Data Acquisition) board.			values that the DAQ is measuring
		D2 (GPIB)	In/Out	Using GPIB connection is stablished the communication of DAQ - PC
System	This section is composed of all the hardware that contain the AMSOP system.	M1 (Step1)	Out	Port to send step values to the driver 1
		M2 (Direction1)	Out	Port to send direction values to the driver 1
		M3 (Step2)	Out	Port to send step values to the driver 2
		M4 (Direction2)	Out	Port to send direction values to the driver 2
		M5 (Step3)	Out	Port to send step values to the driver 3
		M6 (Direction3)	Out	Port to send direction values to the driver 3
		M7 (EndStop1)	In	Port to receive the information of the Endstop switch 1
		M8 (EndStop2)	In	Port to receive the information of the Endstop switch 2
		M9 (EndStop3)	In	Port to receive the information of the Endstop switch 3
		M10 (Serial)	In/Out	The Arduino communicates with LabVIEW application through serial communication

Table 10:BLOCK DESCRIPTION OF THE AMSOP BDD

5.1.1.2 IBD (Interaction Block Diagram)

The IBD of the system is described in the figure below.

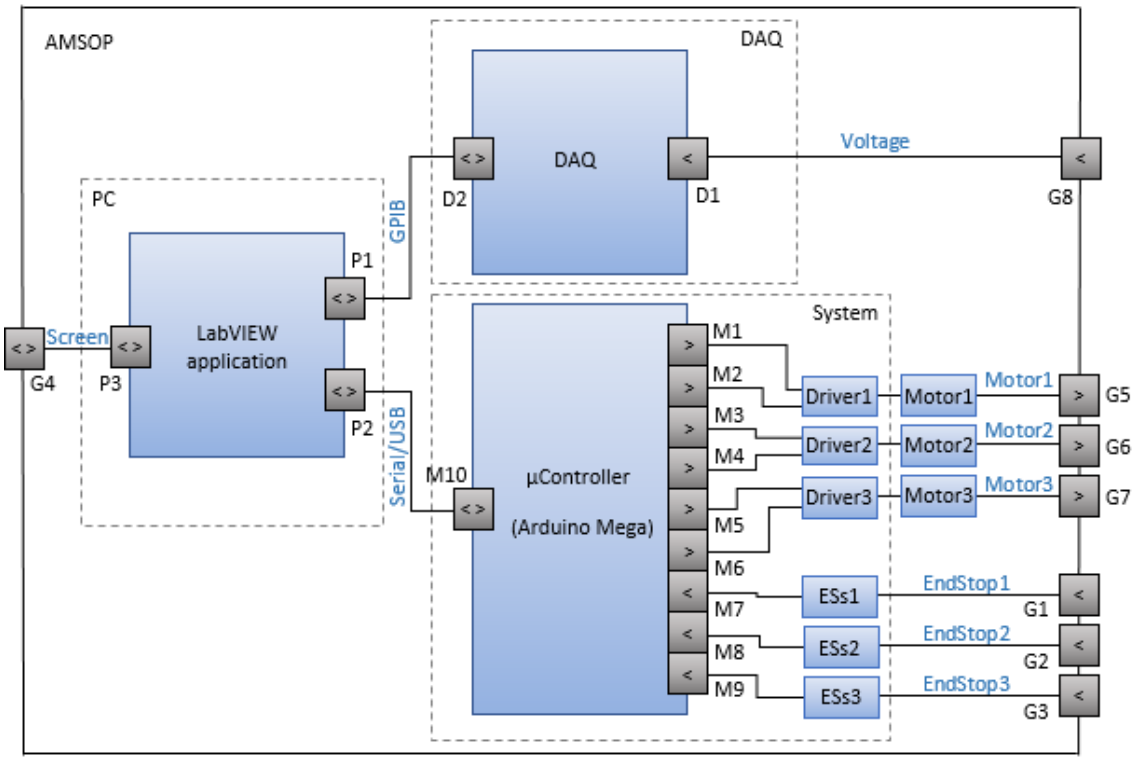


Figure 24:IBD OF THE AMSOP

DESCRIPTION

This table describes each part of the AMSOP that you can find in the IBD of the AMSOP.

Name	Description	Port 1	Port 2
Screen	The user interface will be the way to communication between the AMSOP and the LabVIEW application.	G4	P3
GPIB	By using GPIB communication, it is possible to send and receive data between the PC and the DAQ.	P1	D2
Serial/USB	By using Serial/USB communication, it is possible to send and receive data between the PC and the System.	P2	M10
Voltage	The physical magnitude of the voltage, measured by the DAQ.	D1	G8
Motor1	Signal that is sent by the drivers to move the Motor1.	M1/M2	G5
Motor2	Signal that is sent by the drivers to move the Motor2.	M3/M4	G6
Motor3	Signal that is sent by the drivers to move the Motor3.	M5/M6	G7
EndStop1	Signal that is sent by ESs1 (EndStop switch 1) and this one is received by the Arduino Mega Microcontroller.	M7	G1

EndStop2	Signal that is sent by ESs1 (EndStop switch 2) and this one is received by the Arduino Mega Microcontroller.	M8	G2
EndStop3	Signal that is sent by ESs1 (EndStop switch3) and this one is received by the Arduino Mega Microcontroller.	M9	G3

Table 11:DESCRIPTION PARTS OF THE AMSOP IBD

5.1.2 DAQ

5.1.2.1 BDD

The BDD of the DAQ is described in the figure below.

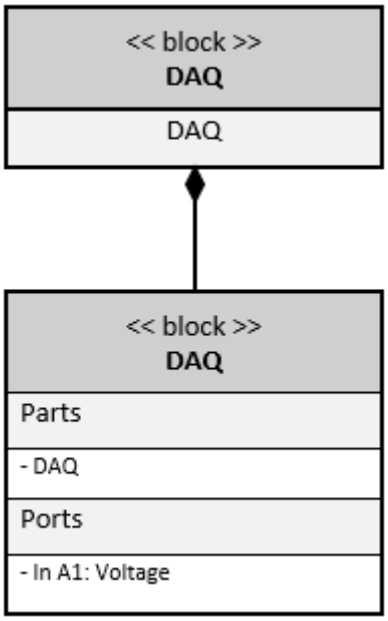


Figure 25:BDD OF THE DAQ

BLOCK DESCRIPTION

In the next table is described the block with their ports.

Block Name	Function Description	Port Name	Direction	Comment
DAQ	The model used to take the voltage data is DAQ NI PCIe-6341	A1	In	Port to receive the voltage data

TABLE 11: Table 12:BLOCK DESCRIPTION OF THE DAQ BDD

5.1.2.2 IBD

On the figure below is the IBD diagram of the DAQ.

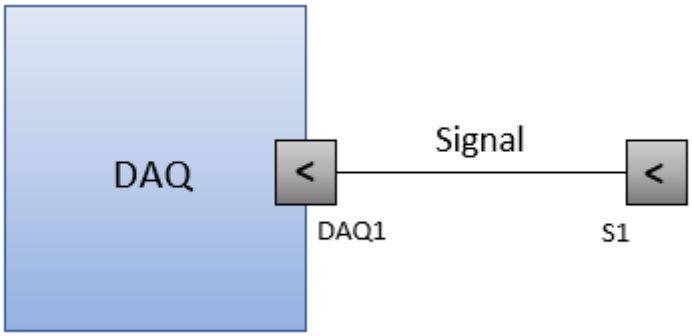


Figure 26:IBD OF THE DAQ

DESCRIPTION

The next table describes DAQ parts where can be seen on the IBD figure.

Name	Description	Port 1	Port 2	Signal Type
Electrical signals	DAQ NI PCIe-6341 is connected through a wire to a conductor end which is going to be in contact with the PCB to know the voltage values	S1	DAQ1	Analog

Table 13:DESCRIPTION OF THE DAQ IBD

5.1.3 System

5.1.3.1 BDD

The BDD of the System is described in the figure below.

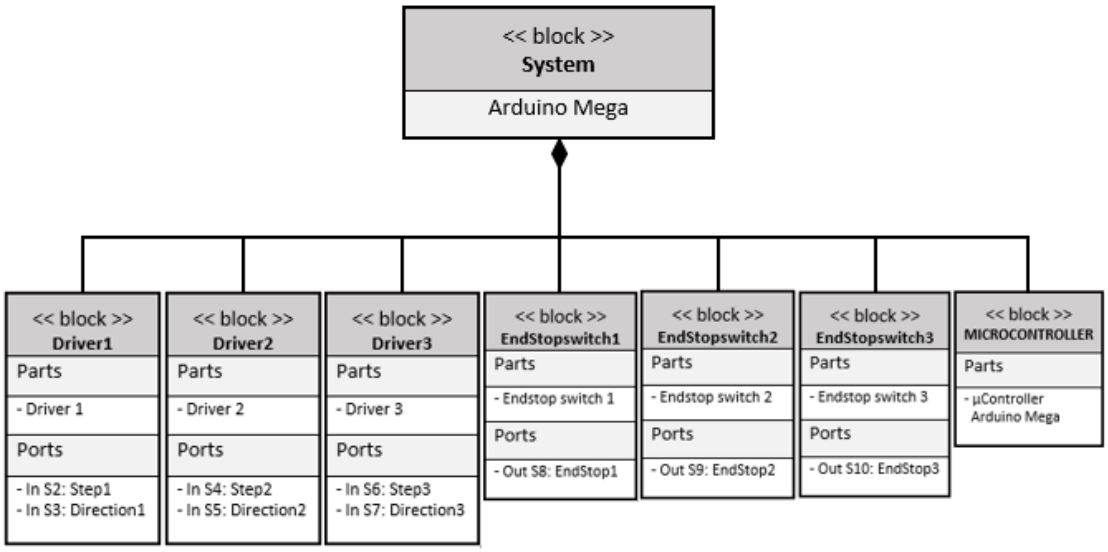


Figure 27:BDD OF THE System

BLOCK DESCRIPTION

In the next table are described all the blocks with their ports.

Block Name	Function Description	Port Name	Direction	Comment
Driver1	The model that is used to control the Motor1 is DRV8825	S2	In	Port to receive the Step1 signal
		S3	In	Port to receive the Direction1 signal
Driver2	The model that is used to control the Motor2 is DRV8825	S4	In	Port to receive the Step2 signal
		S5	In	Port to receive the Direction2 signal
Driver3	The model that is used to control the Motor3 is DRV8825	S6	In	Port to receive the Step3 signal
		S7	In	Port to receive the Direction3 signal
EndStopswitch1	The type of the sensor used of mechanical action	S8	Out	Port to send the EndStop1 signal
EndStopswitch2	The type of the sensor used of mechanical action	S9	Out	Port to send the EndStop2 signal
EndStopswitch3	The type of the sensor used of mechanical action	S10	Out	Port to send the EndStop3 signal
Microcontroller	Controls all components through the LabVIEW application			

Table 14:BLOCK DESCRIPTION OF THE SYSTEM BDD

On the figure below is the IBD diagram of the System.

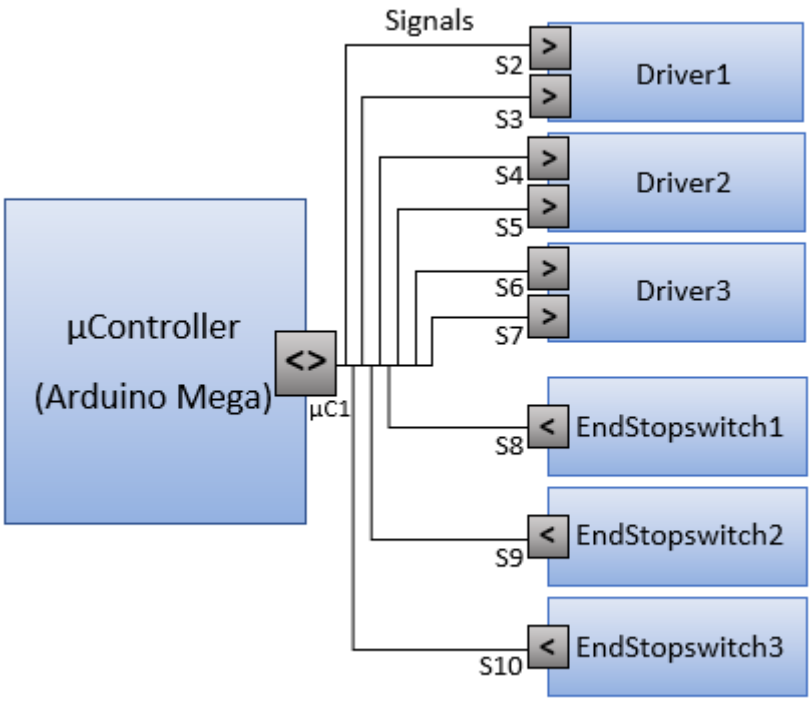


Figure 28:IBD OF THE SYSTEM

DESCRIPTION

The next table describes each part of the System that there are on the IBD figure.

Name	Description	Port 1	Port 2	Signal type
Electrical signals	Three drivers and three endstop switches are connected to the Arduino Mega. Arduino board can get the information coming of the endstop switches to know when the motors are in the limit of the AMSOP and to send the properly information to the drivers to control that motors	µC1	S2	Digital
		µC1	S3	Digital
		µC1	S4	Digital
		µC1	S5	Digital
		µC1	S6	Digital
		µC1	S7	Digital
		S8	µC1	Digital
		S9	µC1	Digital
		S10	µC1	Digital

Table 15:DESCRIPTION OF THE SYSTEM IBDHARDWARE DESIGN

5.1.4 Components

5.1.4.1 Motors

To move all the System, it is necessary use a motor. The differences between the different kind of motors that there are in the market are explained in section "**2. Preliminary analysis** ", so here is commented the parts that has the NEMA 17 stepper motor and how is going to be connected and why.

The motors have two parts:

- Stator: is the part where the coils are. Static part.
- Rotor: is the part that rotates in the motor. It has the magnets.

The idea is that when the coils are magnetized (stator), the rotor is being attracted by the stator.

On the one hand, when the coil is magnetized to the north, the rotor parts of the south are attracted and the motor remains blocked in its position. On the other hand, when the coil is magnetized to the south, the rotor parts of the motor repulse each other and the motor moves a step. That allow us to control the position of the motor and to have an accuracy in the movements.

Due the configuration of the coil, the current can circulate in two directions needing a bipolar control because the rotation direction depends of the current direction, making the movement for one way or the other.

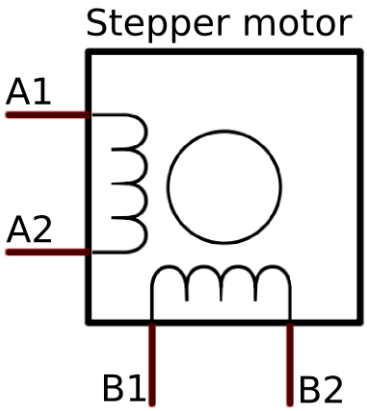


Figure 29:STEPPER MOTOR COILS

Once known how the motor works is needed to know the quantity of current the motor needs.

- Nominal current: current for which the device is designed to work.

This is a specification that the fabricant must say in the datasheet of the motor. This NEMA 17 has a nominal current per phase of 1.7A. That means is needed "something" in order to control the current and not to overheat or damage them. This will be a driver, which is explained in the next section.

The stepper motors have not a continuous movement, then they cannot be in all the angles. The stepper motors that is going to use in this project have 200 steps-per-revolution. This is 360º/200 = 1.8º per step. With this precision is not needed to use any kind of microstepping.

5.1.4.2 Drivers

As before pointed out, the motor coils can be activated with north-south or south-north current. The driver can decide the current direction, changing the polarity of the magnetic field produced inside the motor.

5.1.4.2.1 Composition

To do this, the driver has an H bridge circuit, composed by four transistors which are active in diagonal depending the current direction required.

There are a couple of H bridges in the driver DRV8825, one per motor coil (A and B).

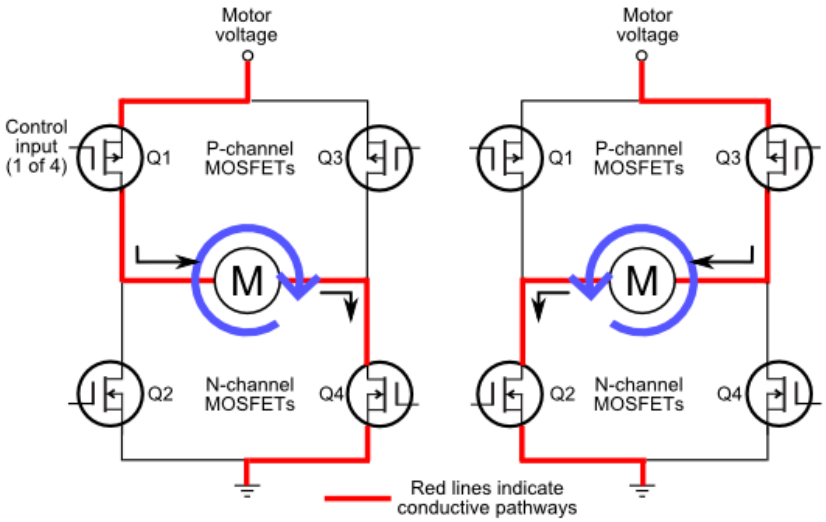


Figure 30:DRIVERS COMPOSITION

5.1.4.2.2 Movement control

The driver is connected to the Arduino Mega using four cables.

Arduino Mega sends the pulse (STEP). Every pulse says to the motor that must take a step.

Another pin says the direction of rotation (DIR). To turn in one direction or the other.

The next pin sends a pulse (FAULT) if steps have been lost.

The last one is ground (GND). Necessary to connect the signals with Arduino Mega.

In this way, every time the driver receives a pulse in STEP pin, the circuit checks the voltage of DIR pin and supply motor coils in the right way.

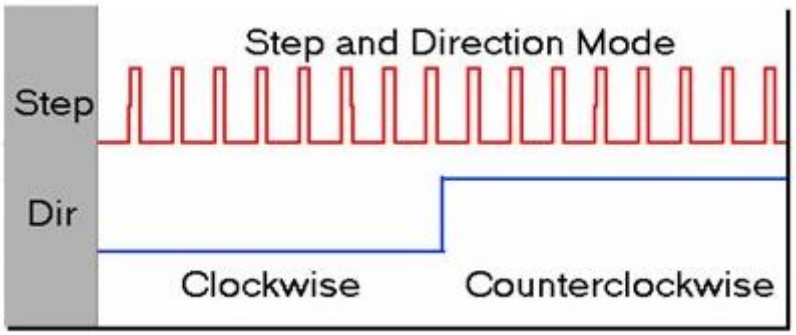


Figure 31:MOVEMENT CONTROL OF THE DRIVERS

5.1.4.2.3 Connection

DRV8825 driver has another pins not connected to Arduino but necessary for its operation. They are described in the table below.

Pin Name	Part	Connection	Pin Description
ENABLE	Digital	Not connected	Allow the driver to send current to the motor
M0	Digital	Not connected	Microstepping configuration
M1	Digital	Not connected	Microstepping configuration
M2	Digital	Not connected	Microstepping configuration
RESET	Digital	SLEEP	Connected to Sleep
SLEEP	Digital	RESET	Connected to Reset
STEP	Digital	Arduino Mega	Indicates that is wanted to take a step
DIR	Digital	Arduino Mega	Indicates motor rotation direction
VMOT*	Analog	Power supply	Motor power supply (between 8V and 45V)
GND	Analog	Power supply ground	Power supply ground
B2	Analog	B motor coil	B motor coil
B1	Analog	B motor coil	B motor coil
A1	Analog	A motor coil	A motor coil
A2	Analog	A motor coil	A motor coil
FAULT	Analog	Power supply	Send a pulse if a step has been lost
GND	Analog	Power supply ground	Power supply ground

Table 16:DRIVER CONNECTION

*It is required a 100 µF capacitor to prevent peaks that could damage the driver (between VMOT and GND).

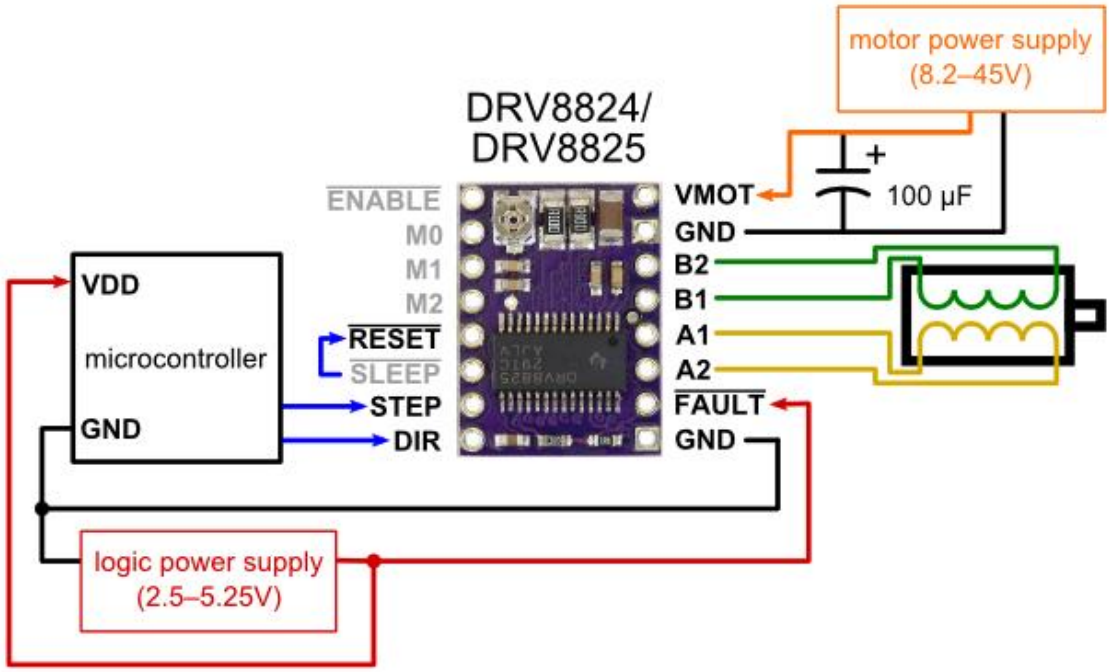


Figure 32:DRV8825 PINOUT

5.1.4.2.4 Current regulation

It is necessary to adjust the current supplied by the DRV8825 drivers, so that, it needs to be calibrated the potentiometer to an 70% intensity of the nominal current per phase of the motor (in our case 1.7A per phase).

Therefore, a motor of 1.7A (nominal current) it will be calibrated to a 1-1.2A of current (1.1A to prevent errors).

To calculate the current is going to use the Ohm's law which says:

$$\mathbf{I = V * R}$$

Every driver has a different voltage-current conversion. In our case (provided by the fabricant):

$$\mathbf{I = V * 2}$$

Doing the equation:

$$\mathbf{V} = I / 2 = 1.1 / 2 = \mathbf{0.55V}$$

It is possible to adjust the voltage in a specific point Vref (potentiometer) making easier to control the current.

5.1.4.3 Endstop switches

There are used three endstop switches in this project, every one of them to detect when in the same axis is found in the position 0 (it is called home position).

The endstop switch is NO (normally open), that means if the switch is not pressed the circuit is going to be open (the current is not flowing). These mechanical endstop switches used in this project are going to be connected to Arduino through a pull-down resistor which causes the pin never to be in the air avoiding possible faults. Pull-down resistor uses the following logic:

- Endstop switched: current flows through the pin → State = HIGH
- Endstop not switched: current passes from the pin to ground (GND) → State = LOW

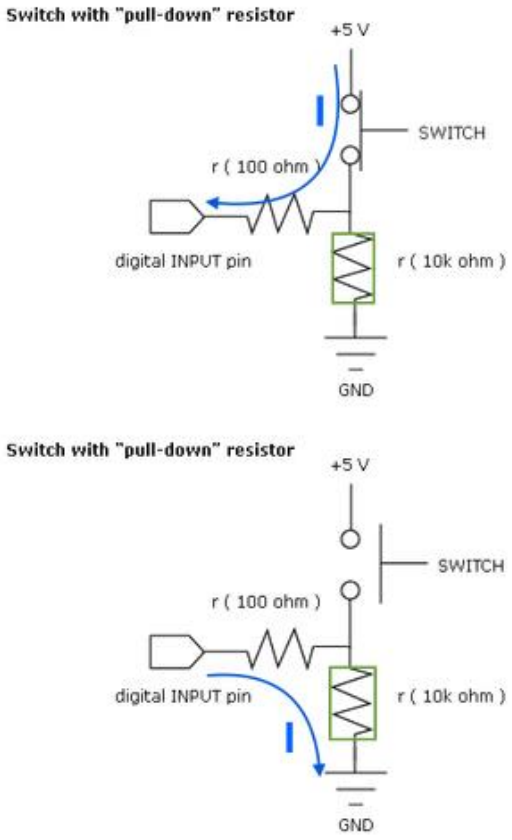


Figure 33:PULL-DOWN RESISTOR

As can be seen from figure above, endstop switch is going to be connected on the one hand, by a voltage of 5V and, on the other hand, by any Arduino digital pin (when it receives HIGH state it known that it has arrived to HOME position) and ground (GND).

5.1.5 Power supply

To supply all the System is necessary an external power supply because neither the PC nor Arduino Mega can supply the motors (they need between 8V and 45V). For that reason, and as have been explained before, an external PC ATX power supply have been modified to do that.

The power supply has a special ATX connector:

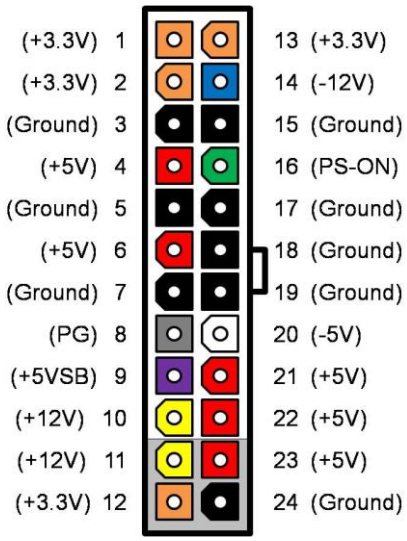


Figure 34:ATX CONNECTOR PINOUT

This connector must be removed and join all the same value cables (in this project have been used 12V and 5V) to make the best possible use of the power.

The only drawback to using this power supply is that the device has been made to supply a PC so, to make it work, the green cable (PS-ON) must be connected to ground (any Ground) to trick the power supply thinking it is working on a PC (strip the cables and put them together).

Two LEDs have been placed to inform when the power supply is on and powering.

With the power supply has been possible to supply the current and voltage necessary for all the System.

5.1.6 Circuit schematics

The figure below (figure 34) is a schematic of the Arduino Mega microcontroller pins.

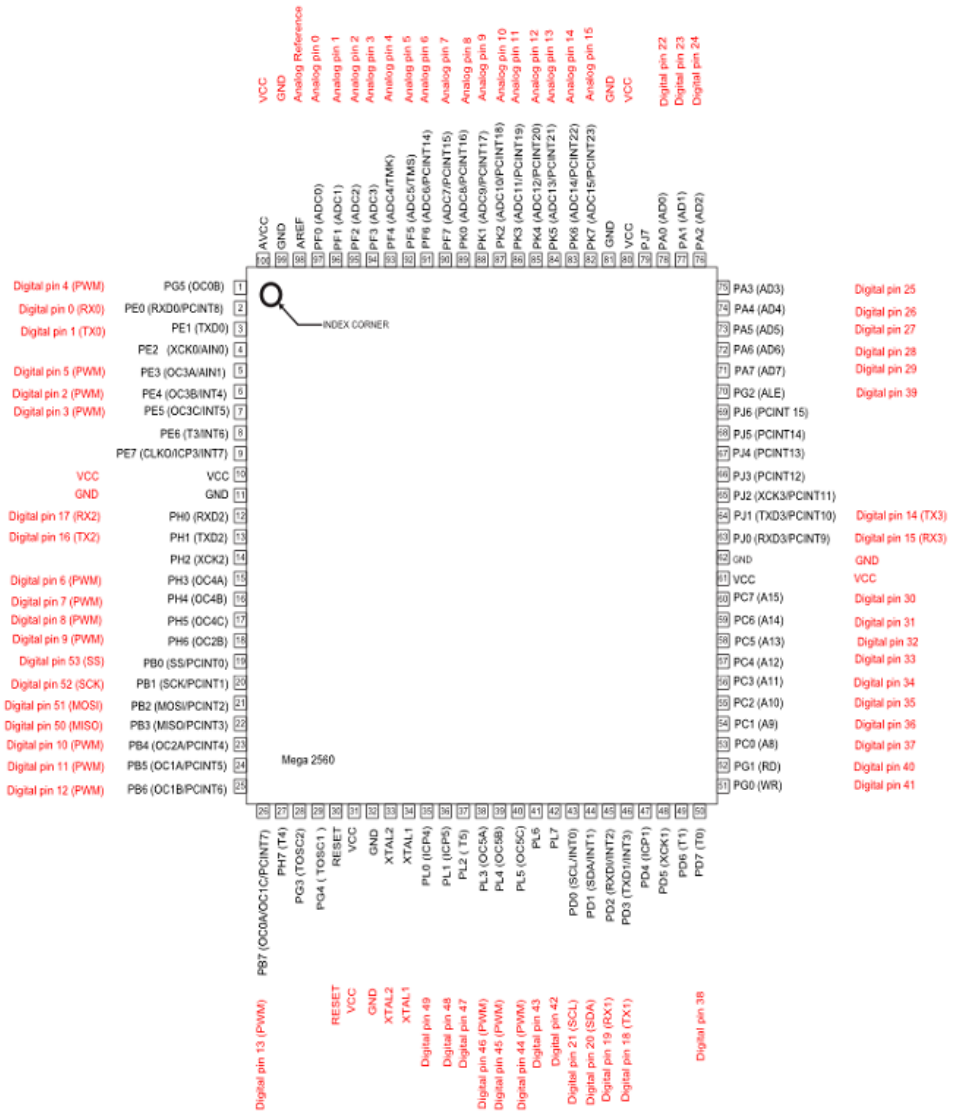


Figure 35:ARDUINO MEGA MICROCONTROLLER PIN DETAILS

On the next page, it will find the schematic of all the system (figure 36). As it is showed below, the alimentation of the System is provided with the 5V PIN of the Arduino Mega board and with 12V and 5V of the external power supply. In the same way, the ground for the system is connected to the GND PIN of the microcontroller and power supply.

On the figure below, you can see too the different devices connected to the Arduino board (Endstop switches, Drivers, Motors) and how they are connected.

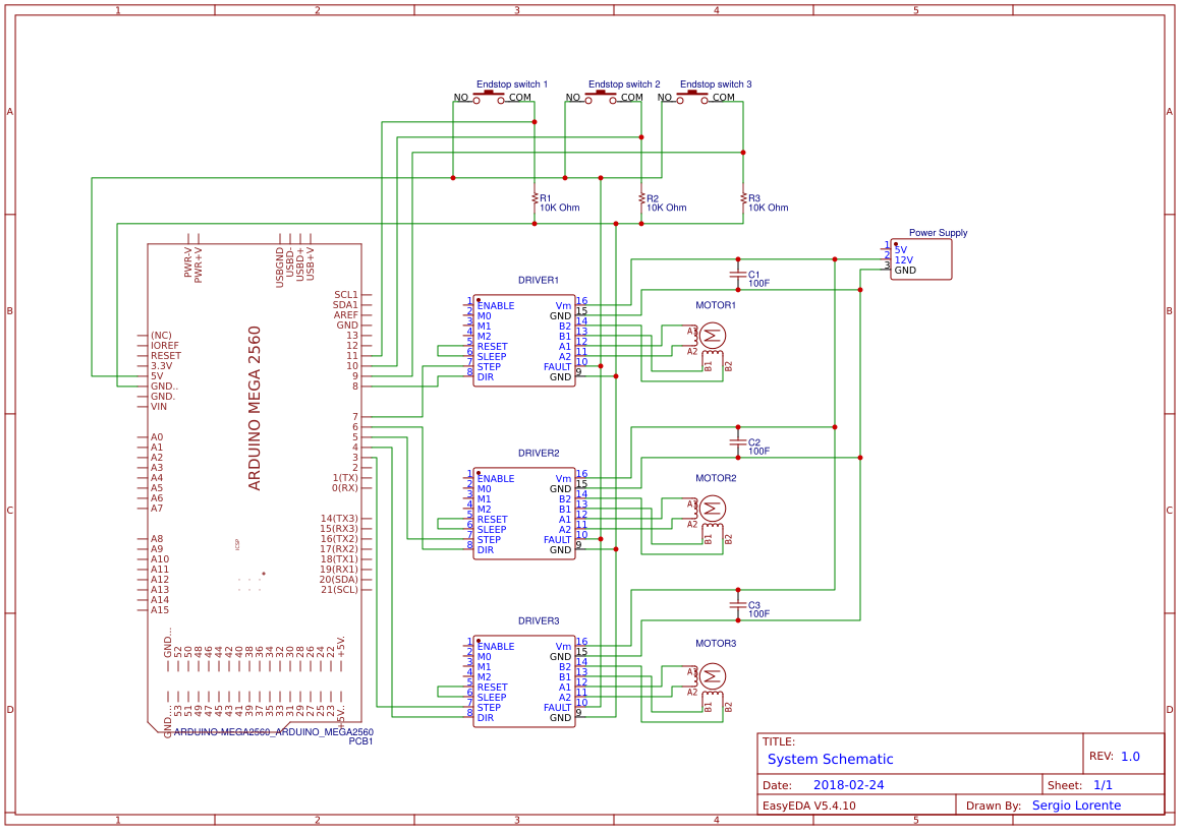


Figure 36:SCHEMATIC AMSOP CONNECTION

Parts of the schematic:

Arduino Mega:

Microcontroller used to control all the System.

Driver1/2/3:

Drivers DRV8825 used in the System in order to limit the current required by the motors (they have to be properly configured). Between Vm and GND is connected a capacitor was added for security, which avoid peaks of voltage that could damage the driver when the power supply is powering. Through the digital pins are controlled DIR and STEP.

Motor1/2/3:

The chosen bipolar motor to move all the System. They need a driver to control them.

Endstop switch 1/2/3:

Mechanical switches that indicate the position home in the System. They need a pull-down resistor to make it work. The resistor is a 10K Ohm connected with COM pin. NO (normally open) pin is supplied with 5V, voltage that indicates HIGH sate.

Power supply:

In the System, the motors need an extra power to make them work. Basically, was incorporated to supply them with 12V.

5.1.7 Hardware implementation

In this point is showed how the AMSOP hardware part was implemented. In the figure below (figure 36) it is possible to see all the parts of the System, the mechanical structure with the motors, the power supply powering the drivers and the drivers and endstop switches connected to Arduino Mega.

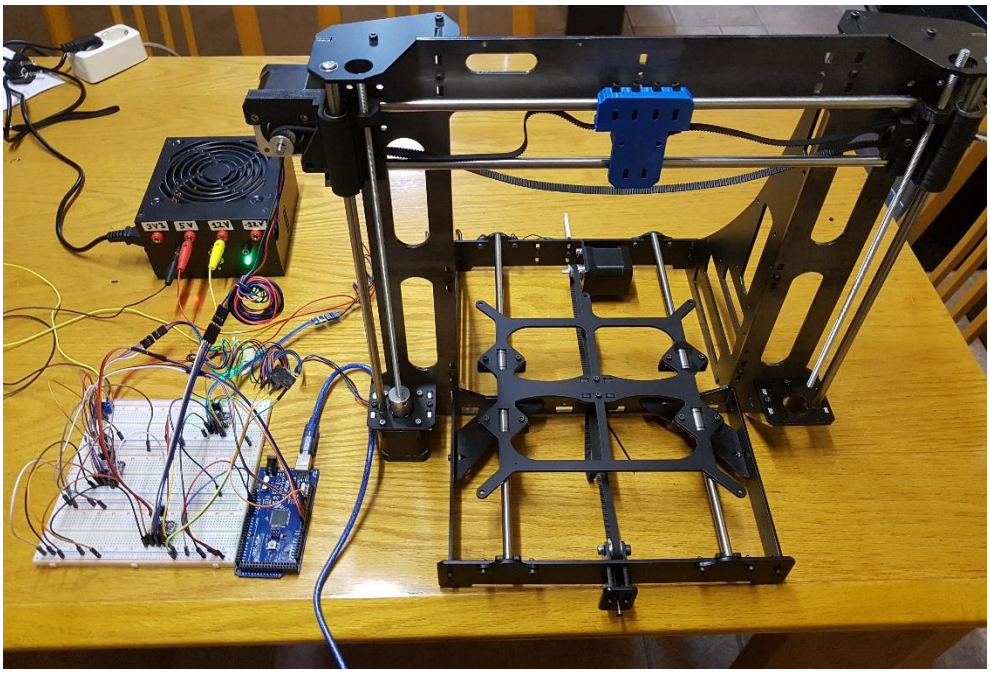


Figure 37:GENERAL VIEW OF THE AMSOP

6 SOFTWARE ARCHITECTURE AND DESIGN

This section aims at describing the software architecture and design of the AMSOP. To do it, is going to be develop the 4 + 1 view model. This kind of model, is based on the use of multiple concurrent views. The views are used to describe the system from the viewpoint of different stakeholders, such as end-users, developers and project managers.

6.1 DOCUMENT STRUCTURE AND READING INSTRUCTIONS

In this document it can be found the description of each following views: this document is an overview of the software architecture.

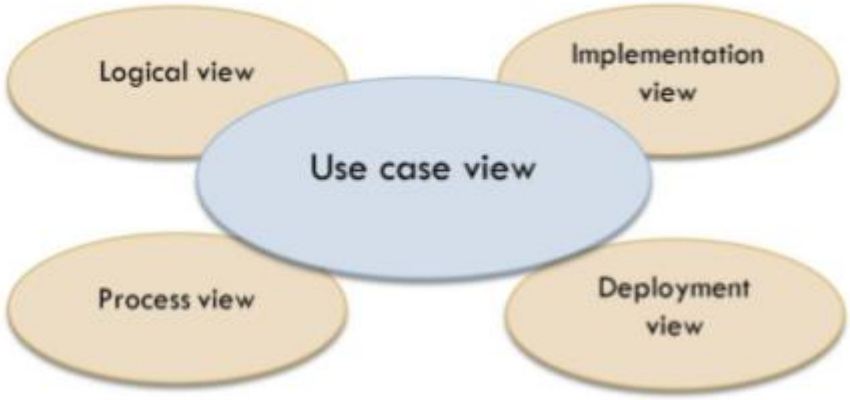


Figure 38:4+1 VIEW MODEL

6.1.1 Use case view

The aim of this view is to regroup all information about the system use. From how the system should be used to how it should work, with some scenario to illustrate it.

6.1.2 Logical view

The purpose of this view is to describe the interaction internal of the AMSOP. The interaction can be observed between the different indicator/controls, structures, functions, states…

6.1.3 Implementation view

This view presents all software components and file structure of the system. Also known as development view.

6.1.4 Process view

The idea of this view is to show every step of every process. With all interactions and threads running, in parallel or one after the other.

6.1.5 Deployment view

This view describes hardware components and how they are implemented with software. Add to this, the different communication protocols used.

6.2 LOGICAL VIEW

6.2.1 LabVIEW application (VI)

In this section, it is described all about the LabVIEW application created for the PC which will be connected to the System and the DAQ. It will describe the indicators/controls, functions, Sub VI, structures… then there will be diagrams about each use case and how it is implemented in a state machine.

6.2.2 Description of states

There are described the different states of the LabVIEW application. The complete states programmed are pasted in the annex [**Annex I**].

Initialize

This is the initial state of the machine, it is where all the constants, arrays and ports are initialized.

Load Coordinates

This state allows the user to upload the coordinates to the program that will decoder the .txt into a table that will be shown in the front panel.

Conversion

This is the state was created to update the coordinates, once a test point is read this state allow to load the new coordinates and show it in the front panel.

XY Movement

In this state, the program sends the signal to the drivers that will make the motors X and Y (1 and 2) move. There are sending two signals, one of them for the DIR and the other one for STEP.

Plot Properties

A new picture of the current state of the test points is created in the front panel view. It is updated the “picture” of the PCB.

Z Movement

As the XY Movement state, Z Movement state will manage the Z motor sending two digital signals as well, DIR and STEP.

Data Acquisition

This state will acquire the signals of the PCB test points. It is sending the order to read voltage to the DAQ. The graphic is upgraded with each sample.

Home

Home state was created to supply the need to know the initial position of the axis. In this state the motors will turn until the endstops switches signal will be detected.

Save Data

The state will make and save the report with the measures taken before.

Exit

This state allows to close the program and close all the stablished communications.

6.2.3 Use case 1: Connect System – LabVIEW application

Get a connection between the System and the application.

· Pre-condition: The PC and the System are connected.

· Post-conditions: The system and the application are connected.

6.2.3.1 States involved

The following states (see figure 39) were involved for that use case:

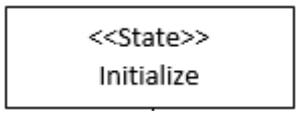


Figure 39:STATES INVOLVED IN UC1

6.2.3.2 Sequence diagram

Connecting the System to LabVIEW application consists in that the user stablishes a communication between PC and Arduino through a USB cable. Then, click run button. This sequence diagram (see figure 40) describes how the user can connect the System to the application by using USB cable.

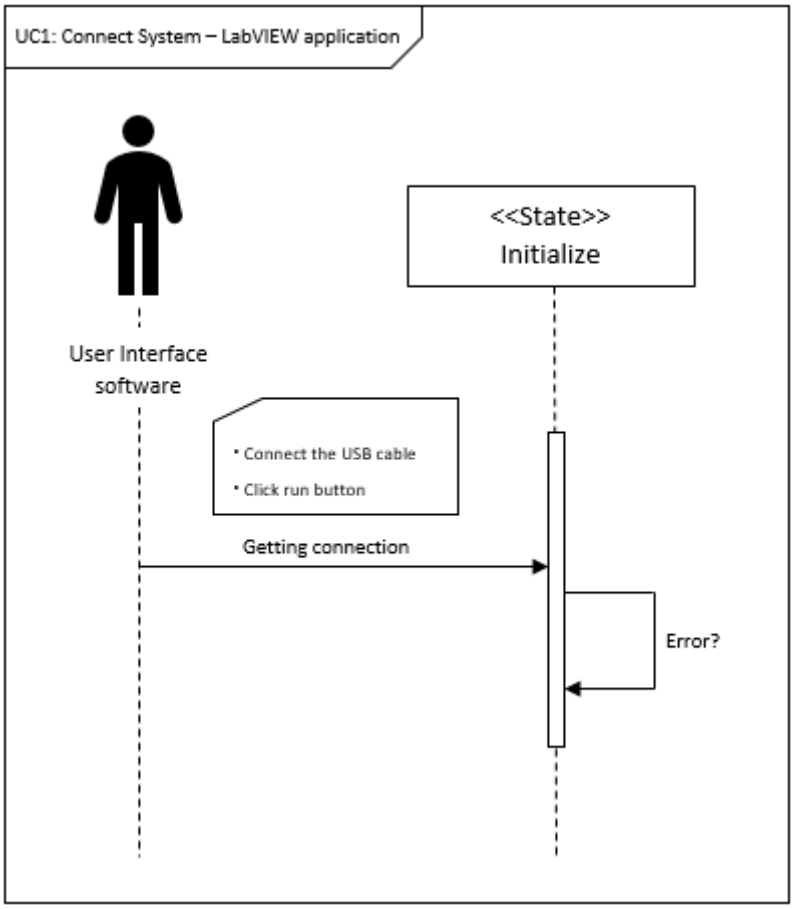


Figure 40:SEQUENCE DIAGRAM OF THE UC1

6.2.4 Use case 2: To load the program

The user loads the coordinate program in the application.

· Pre-condition: The application is running.

· Post-condition: The coordinate program is loaded.

6.2.4.1 States involved

The following states (see figure 41) were involved for that use case:

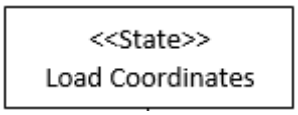


Figure 41:STATES INVOLVED IN THE UC2

6.2.4.2 Sequence diagram

Loading the coordinates consists in that the user selects the relevant coordinate program to upload in the application. This sequence diagram (see figure 42) describes how to load the program.

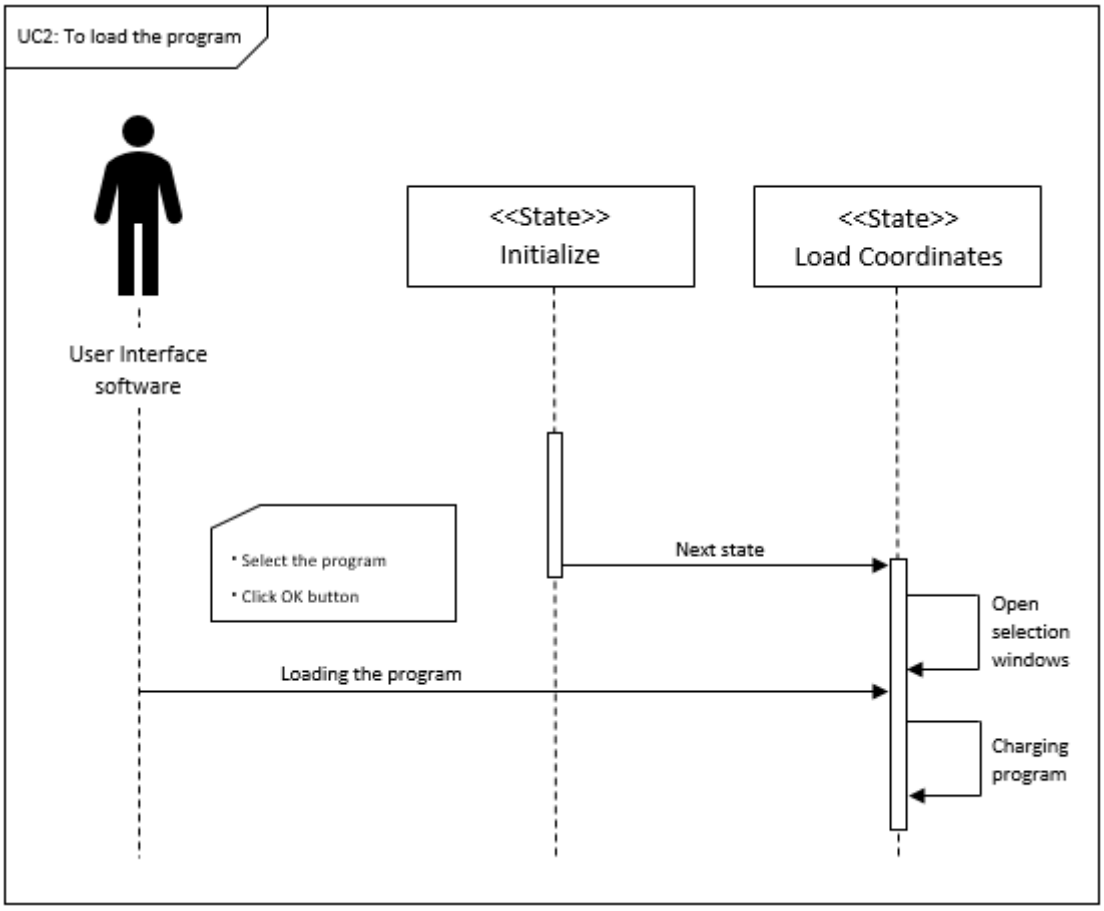


Figure 42:SEQUENCE DIAGRAM OF THE UC2

6.2.5 Use case 3: Starting and stopping

The user can easily start and stop the AMSOP program.

· Pre-condition: The application is running and the system is correctly powered.

· Post-condition: The AMSOP is correctly started and stopped.

6.2.5.1 States involved

The following states (see figure 43) were involved for that use case:

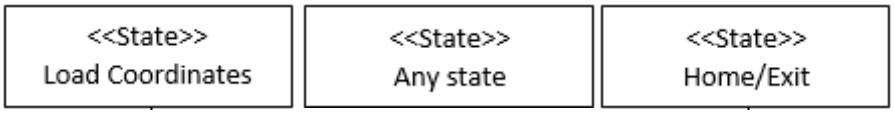


Figure 43:STATES INVOLVED IN THE UC2

6.2.5.2 Sequence diagram

Starting and stopping consists in that the user toggle a button and the program must recognize if start or stop. This sequence diagram (see figure 44) describes how the user can start or stop the program.

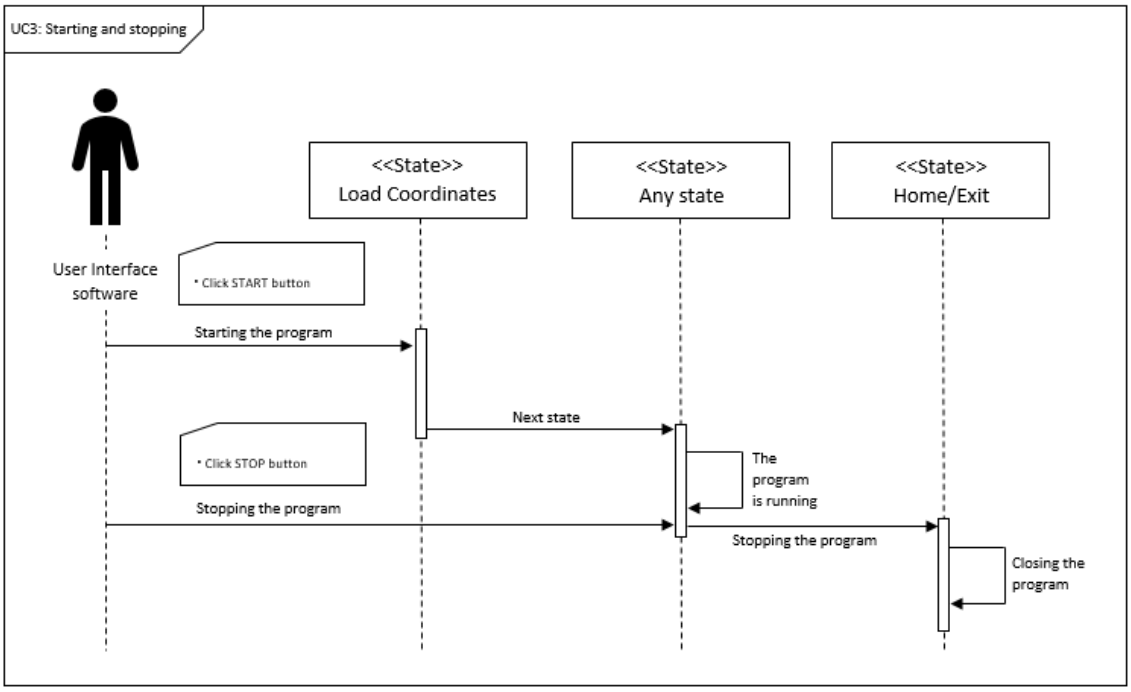


Figure 44:SEQUENCE DIAGRAM OF THE UC3

6.2.6 Use case 4: Receive alerts

The user gets the information of the issue in the screen.

· Pre-condition: The application is running.

· Post-condition: The alerts are displayed on the screen.

6.2.6.1 States involved

The following states (see figure 45) were involved for that use case:

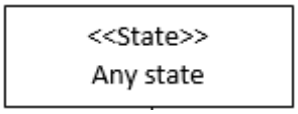


Figure 45:STATES INVOLVED IN THE UC4

6.2.6.2 Sequence diagram

Receiving alerts consists in that the user can see in the screen when an error has occurred. This sequence diagram (see figure 46) describes how the user can receive alerts.

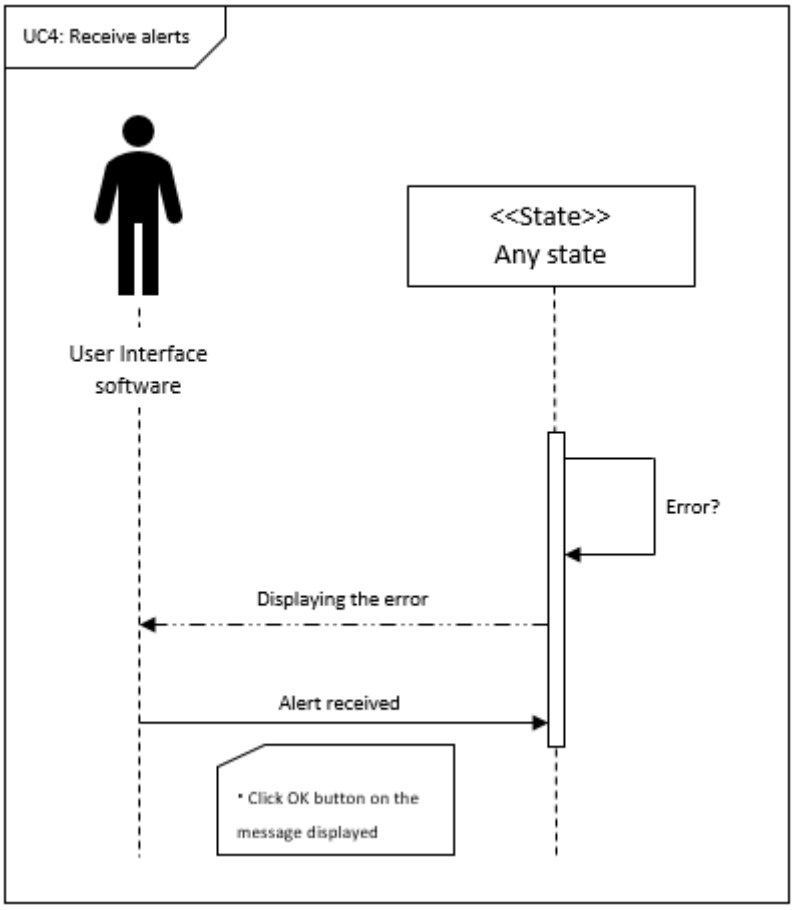


Figure 46:SEQUENCE DIAGRAM OF THE UC4

6.2.7 Use case 5: Data processing

The data is being processed

· Pre-condition: The application is running and the system is correctly powered.

· Post-condition: The data is correctly processed.

6.2.7.1 States involved

The following states (see figure 47) were involved for that use case:

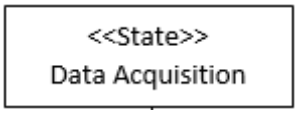


Figure 47:CLASSES INVOLVED IN THE UC5

6.2.7.2 Sequence diagram

Data processing consists in that the program starts to take the samples and saving them. This sequence diagram (see figure 48) describes how the program is taking the data and it is displayed in real time.

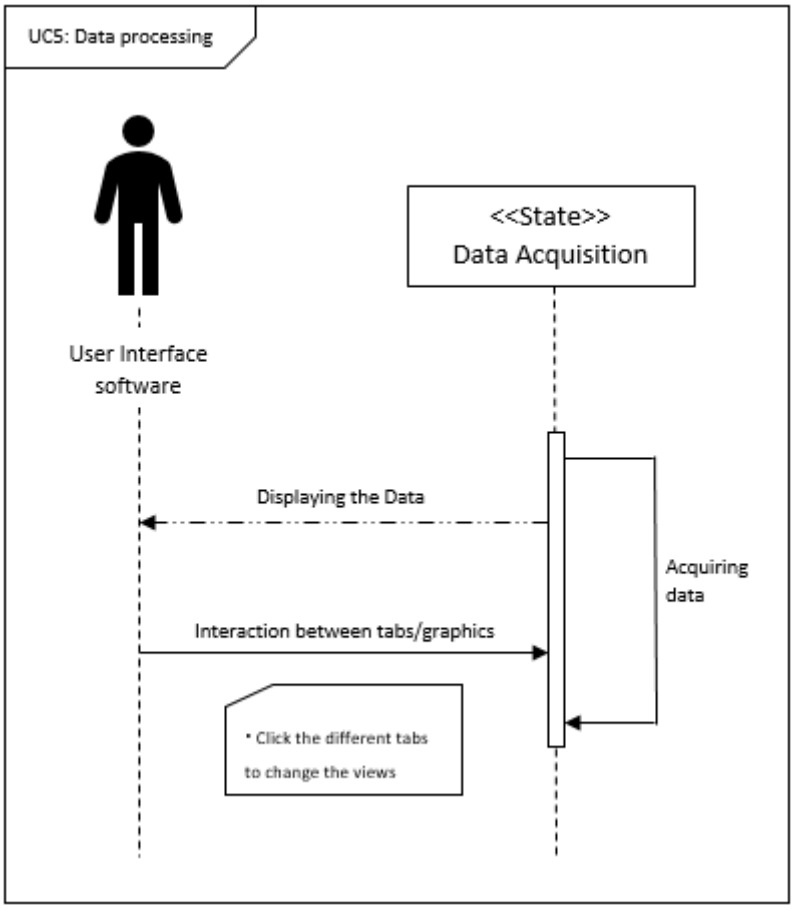


Figure 48:SEQUENCE DIAGRAM OF THE UC5

6.2.8 State diagram

The figure 49 shows the complete state machine diagram of the LabVIEW application used on the PC. Shows all the implemented states and how are interrelated.

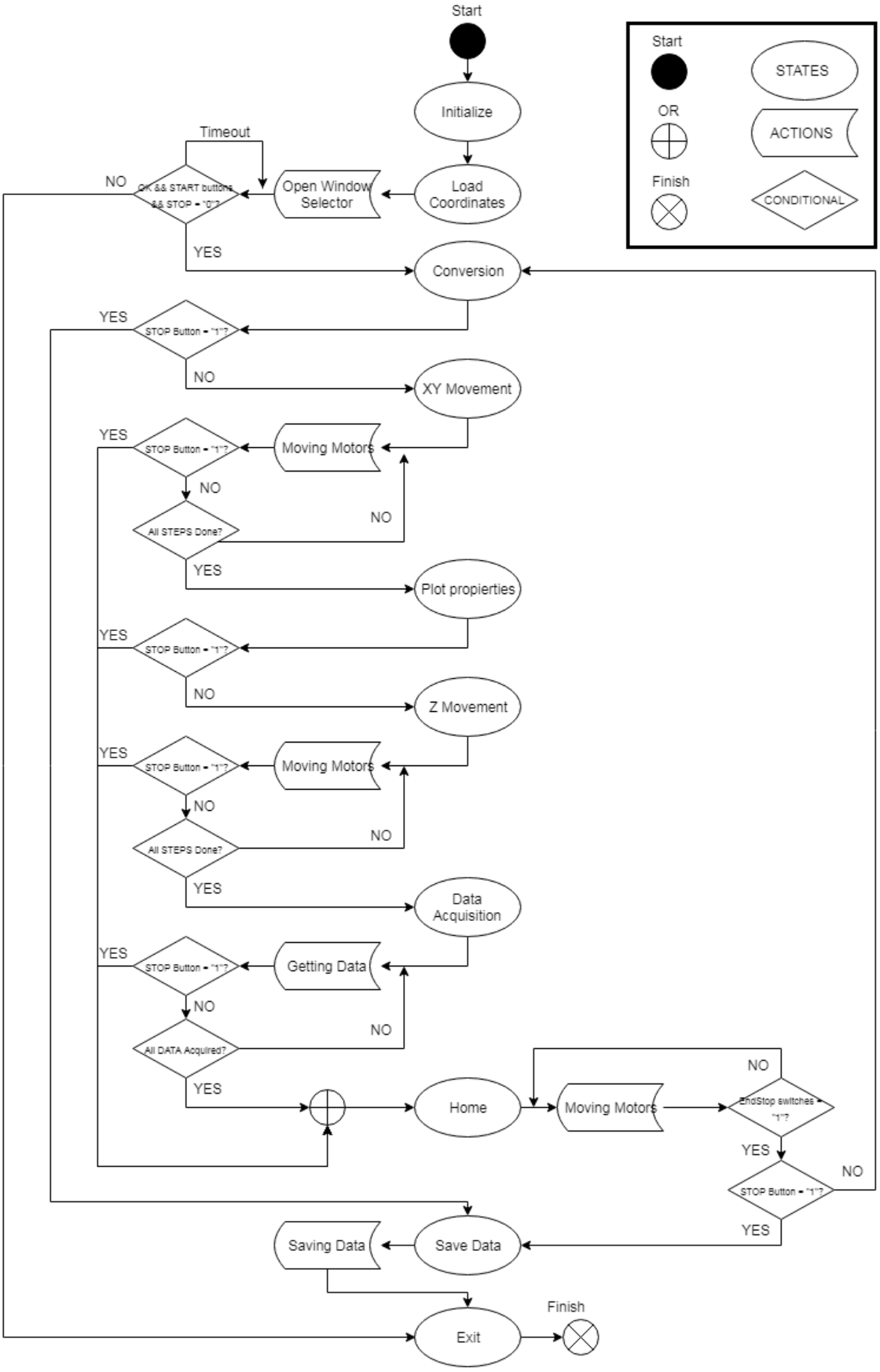


Figure 49:STATE DIAGRAM OF THE LABVIEW APPLICATION

6.3 DEPLOYEMENT VIEW

The deployment view shows the hardware environment in which the AMSOP has to be executed. This view visualizes the physical parts where software is deployed. In this section it is shown the nodes and how these nodes relate between each other.

6.3.1 Overview

The figure below shows how the System communicates with the LabVIEW application.

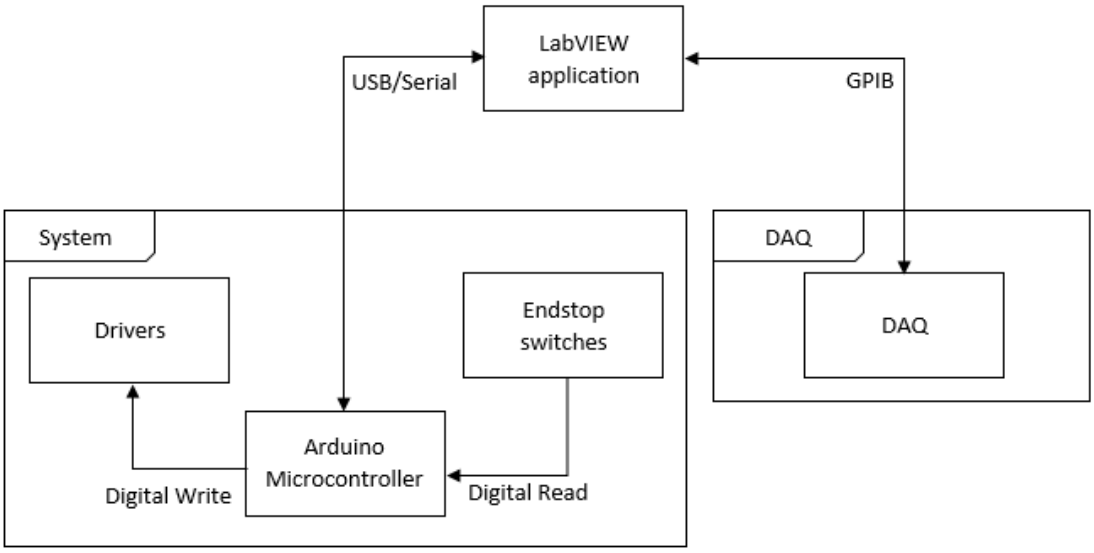


Figure 50:DEPLOYMENT DIAGRAM

6.3.2 Node description

LabVIEW application

With the LabVIEW application the user can access to control the System and the DAQ. On the diagram above it is seen that in order to reach the System microcontroller, all the components values and ability to move the motors, first the access should be made through USB/Serial communication. To take the values it is connected through GPIB to the DAQ.

System

The System is a complex node which involves smaller nodes i.e. Arduino microcontrollers, drivers and endstop switches. The main component of the System node is the microcontroller as every other part of the greenhouse system are connected to it.

DAQ

The DAQ is the other important part because it takes the samples of the board under test. The data is send to the PC via GPIB and the application gets this data and displays to the user

6.3.3 Protocol

USB/Serial Read

The components are connected to the Arduino board and to read and write the data it is used the serial communication. The baud rate is 9600 bits/second. The Serial command stablishes the communication with the Arduino that reads or write from and to the components which are later sent or received by the LabVIEW application.

GPIB

The DAQ is connected to the PC through GPIB communication, this standard allows the DAQ to be communicated with the PC. The information is decoded by the board and sent to the application.

Digital Write

The drivers are connected to the digital pins in Arduino, so it has only two states – HIGH or LOW. With a command it is possible to send the signals.

Digital Read

Similar to digital write, the endstop switches are connected to the digital pins in Arduino. With a command it is possible to receive the signals.

6.4 PROCESS VIEW

The process view section describes the interaction between all the components of the AMSOP. Main components are the LabVIEW application and the System (the Arduino board).

6.4.1 LabVIEW application

The LabVIEW application software has been used as the control for the AMSOP. Inside the LabVIEW application was used different tasks:

- UI Thread (Main thread)
- States (Thread)

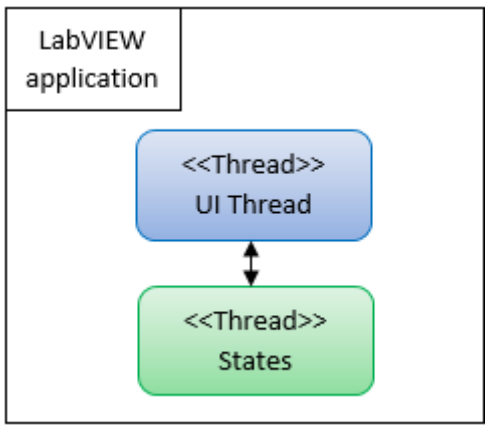


Figure 51:LABVIEW APPLICATION THREADS

To receive and sending data from the components, it is needed to connect the microcontroller. For that, it is used States thread. It contains the request which allows the application to get the data from the Arduino

6.4.2 System

The other big component is the "System". The System contains a microcontroller. The microcontroller it is going to use is ATmega2560 that it is integrated in Arduino Mega. It doesn't support multithreading so there'll be only one thread. Was used it to receive information of the components and control them.

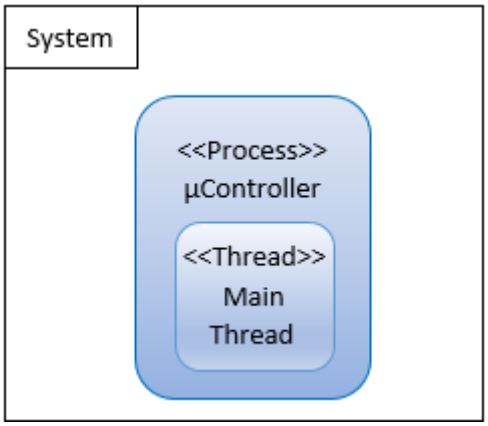


Figure 52:SYSTEM THREADS

6.4.3 Process interaction

The following figure shows how the threads are related between them.

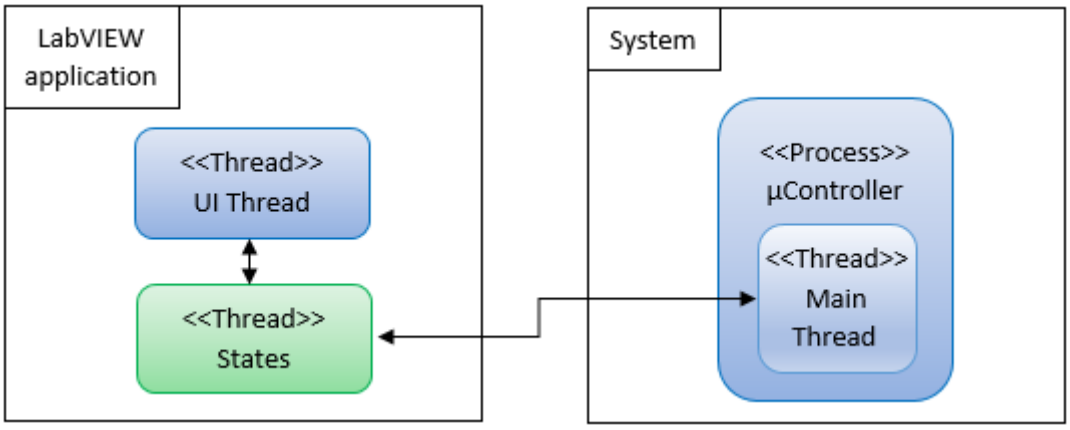


Figure 53:PROCESS INTERACTION

6.5 IMPLEMENTATION VIEW

The implementation view describes the software component structure. In this section it will be described the software tools required to program the microcontroller and design the LabVIEW application.

NI LabVIEW 2017

To create the LabVIEW application, it had been used NI LabVIEW 2017 as it is straight forward, easy in layout creation and programming. When the program is uploaded to the LabVIEW application (Vi) it is easy to test it, find the mistakes, debug and fix the errors.

Arduino IDE

To program the Arduino microcontroller, it is reasonable to use Arduino IDE software. Due to all the AMSOP (Arduino included) is controlled by the LabVIEW application, it was installed a predefined firmware by LINX in Arduino.

6.5.1 Software implementation

The following is a more detailed explanation about the actions taken, seen in the state machine diagram, explaining how they were implemented.

6.5.1.1 Open window selector

As was explained above, the action of to open a window selector is implemented in the code of the LabVIEW application. Basically, the "Load Coordinates" state uses "Read from text file" function to open a prompt and allow the user to explore the PC and select the appropriate program to load it.

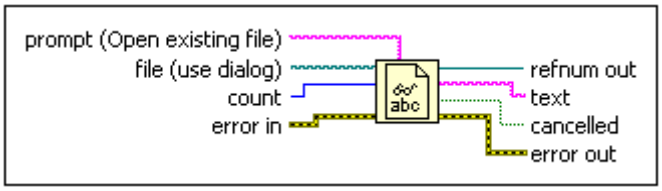


Figure 54:READ FROM TEXT FILE FUNCTION

6.5.1.2 Moving motors

In all cases where is needed to move the motors, the principal functions used are "Digital write" and "Digital read". These functions allow write and receive values to and from a specific output/input channel.

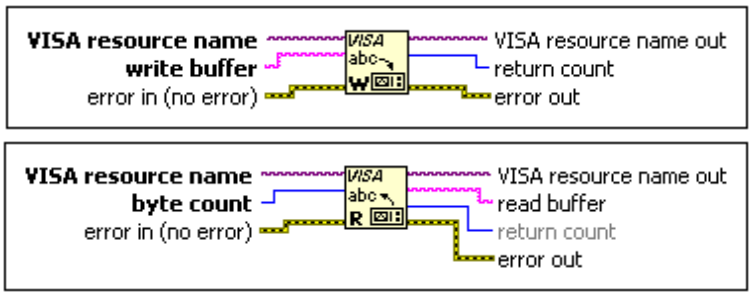


Figure 55:DIGITAL WRITE AND DIGITAL READ FUNCTIONS

The Digital write.vi has been used because the function digital write is already implemented inside this .vi. The same use is given to Digital read.vi, where the function implemented is digital read. Furthermore, digital write.vi allows linking a LINX Resource (Arduino board) and selecting a specific pin to give it value (HIGH or LOW).

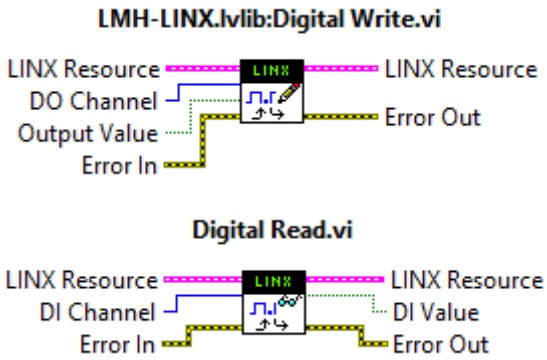


Figure 56:DIGITAL WRITE.VI AND DIGITAL READ.VI

On the one hand, Digital write .vi are used to select the correct pins of the Arduino to send the values “1” and “0” to STEP and DIR of the drivers. Give to STEP pin a “1” means take a STEP and give “1” or “0” to the DIR pin of the driver means turning left or right the motor.

On the other hand, Digital read.vi is used to check the state of the endstop switches, where, if is receiving a “1” means the endstop switch has been switched.

6.5.1.3 Getting data

Once the motors have been correctly positioning, the program will be in the “Data Acquisition” state, and it could start making samples. To do it, is used the DAQ Assistant, included with NI-DAQmx, that is a graphical, interactive guide for configuring, testing, and acquiring measurement data. With a single click, it is possible to generate code based on the configuration, making it easier and faster to develop complex operations.

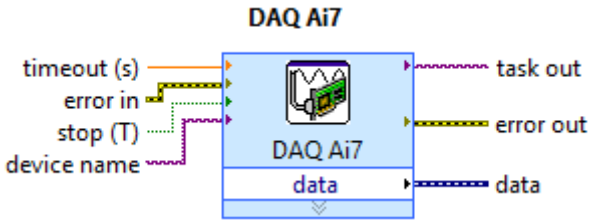


Figure 57:DAQ ASSISTANT

The DAQ Assistant has been configured to read the analog values from a specific port (in this case, Ai7) when is desired.

6.5.1.4 Saving data

The last big action is to save the data taken by the DAQ Assistant. To create an Excel file is really easy to do with write to measurement file, that writes data to Microsoft Excel files (.xlsx) between others being correctly configured.

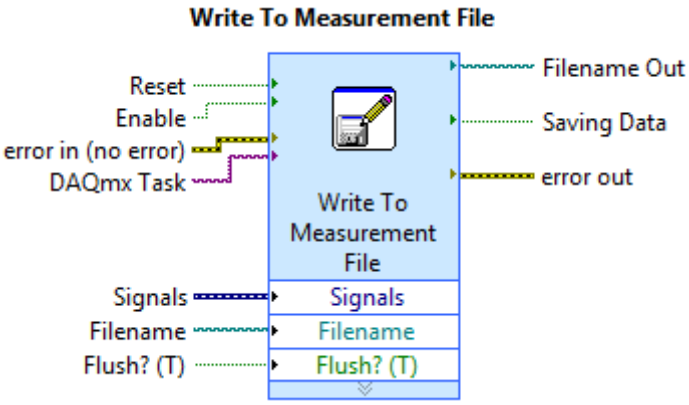


Figure 58:WRITE TO MEASUREMENT FILE

With this output express VI is possible to save it where, how and with the wanted name. It is the perfect tool to generate the final report.

7 ACCEPTANCE TEST

7.1 PURPOSE

The purpose of the section is to define the acceptance test for the AMSOP.

7.2 DEFINITIONS

The "User's PC" refers to the PC where the LabVIEW application is installed.

The "AMSOP System" refers the to the AMSOP with the motors, endstop switches, drivers and the microcontroller.

7.3 SCOPE

This acceptance test checks requirements written on requirement specification.

7.4 TEST SPECIFICATION

The test specification specifies which devices are tested and in what conditions the tests are performed.

7.4.1 Devices under test (DUT)

These devices are part of the AMSOP System:

Devices	Operating System	Hardware
Microcontroller	N/A	Arduino Mega
Driver	N/A	DRV8825
Endstop switch	N/A	Mechanical switch
Motor	N/A	Nema17
User's PC	Windows 10	PC

Table 17:LIST OF TESTED DEVICES

7.5 TEST SETUP

The AMSOP System consist of drivers, endstop switches, motors and Arduino Mega board. It will make the proof of concept connecting the Arduino board to the PC through the USB cable in order to communicate and transfer the data.

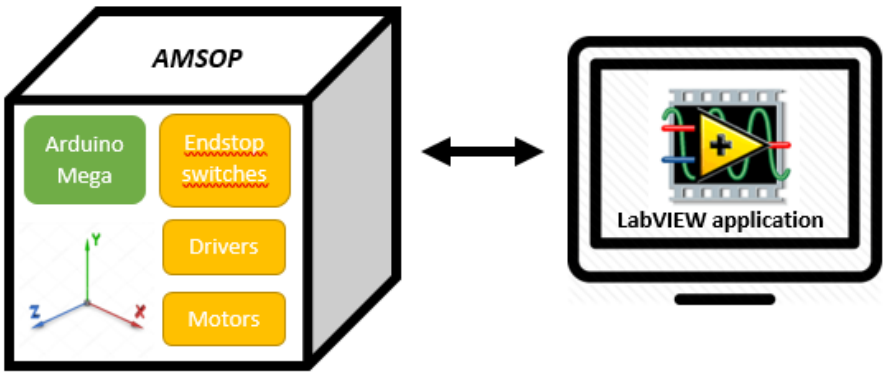


Figure 59:TEST SETUP

7.5.1 Test Environment

The AMSOP System and the User's PC should be close. The range depends on the length of the USB cable (approximately 1 meter). Also, the System and the PC are required to be powered.

7.6 UNIT TEST

7.6.1 Testing LabVIEW – Arduino communication

The communication test has been made trying to control an Arduino Mega LED from the LabVIEW application. To do that, it has been sent a "HIGH" and "LOW" digital values to pin number 13 of Arduino which has an integrated LED. Finally, the LED started to blink.

Test Name	Test Result
Testing LabVIEW – Arduino communication	Passed ✓

Table 18:COMMUNICATION UT

7.6.2 Moving motor

The motors are controlled by the drivers, so, to move them it is necessary to check the drivers as well. It has been checked if when it is send a pulse to the driver digital pin STEP, the motor makes a step and changing the DIR digital pin value the motor starts to change the direction of rotation. Properly configured, the motor starts to work correctly.

Test Name	Test Result
Moving motor	Passed ✓

Table 19:MOVING MOTOR UT

7.6.3 Testing endstop switch

To test the endstop switches, it has been sent digital read command to Arduino in the specific pin of the endstop switch signal (remember that was configured with a pull-down resistor) and when the switch is pressed, the value of the digital pin goes to "HIGH".

Test Name	Test Result
Testing endstop switch	Passed ✓

Table 20:ENDSTOP SWITCH UT

7.7 INTEGRATION TEST

The following test describes the experiments where the separate parts, tested in the unit test, are combined. This test is very important to perform as it shows the functioning of the full system and the possible errors.

7.7.1 Combine all components

Basically, this test is trying to combine with the communication between the LabVIEW application and Arduino, the movement of the motors and how they stop when the endstop switches are pressed. A specific program to make this test has been made which shows that the communication between the components is correct.

Test Name	Test Result
Combining all components	Passed ✓

Table 21:ALL COMPONENTS TEST

7.8 TEST PROCEDURE

7.8.1 Functional requirements test

7.8.1.1 Use case 1: Connect System – LabVIEW application

Description: The test checks whether the connection is established between the System and the LabVIEW application.

Precondition: · The PC and the System are correctly powered.

· The PC and the System are running.

· The PC and the System are in a close area.

Step	Procedure	Expected Result	Test Result
1	Connect the USB/Serial cable from Arduino to PC	Arduino's power LED is on.	Passed ✓
2	Upload the firmware in Arduino from LabVIEW	Arduino's Rx and Tx LEDs are blinking.	Passed ✓
3	Open the LabVIEW application and press "RUN" button.	The LabVIEW application shows the procedure message with no errors.	Passed ✓

Table 22:UC 1 TEST

7.8.1.2 Use case 2: To load the program

Description: The test performs the load of the coordinate program used to mark the test points.

Precondition: · The application is running.

Step	Procedure	Expected Result	Test Result
1	Select the corresponding program	A selection window is displayed and is possible to select the program	Passed ✔
2	Click the "OK" button to load the program	The program is loaded and the front panel is updated (Test points and PCB image is displayed)	Passed ✔

Table 23:UC 2 TEST

7.8.1.3 Use case 3: Starting and stopping

Description: The test checks the capability of the AMSOP to move the motors, translate the movement in the plot and receive the answer of the endstop switches, as well as closing the communication.

Precondition: · The PC and the System are correctly powered.

· The PC and the System are running.

· All components of the System should be connected to the correct pins.

Step	Procedure	Expected Result	Test Result
1	Press the "START" button	Check the how the state of the LabVIEW application is changing	Passed ✔
	Check if the motors are in movement	The motors should be in movement in the movement states	Not Passed ✖ (The Z motor has not enough force)
	Check if the plot is being uploaded in the movement states	The image plot must change simulating the motors movement	Passed ✔
	Check that the signal of the endstop switches is correctly received when are pressed	The movement of the motors must stop	Passed ✔

2	Press the “STOP” button	Check that the application is being stopped and the communication is closed	Passed ✔

Table 24:UC 3 TEST

7.8.1.4 Use case 4: Receive alerts

Description: The test performs the display of alerts on the LabVIEW application.

Precondition: · The PC and the System are correctly powered.

· The PC and the System are running.

Step	Procedure	Expected Result	Test Result
1	Disconnect the USB cable and click “RUN” button to simulate a mal connection failure	A message is displayed indicating the problem	Passed ✔
2	Try to load a non-valid format of program to the application	A message is displayed indicating the invalid format	Passed ✔
3	Change the path of saving the report to an invalid one	A message is displayed indicating invalid path to save it	Passed ✔

Table 25:UC 4 TEST

7.8.1.5 Use case 5: Data processing

Description: The test checks the capability of the AMSOP to display the data of the DAQ and generate a report.

Precondition: · The application is running.

· The pin of the DAQ must be the correct one.

· The path to save the report must be valid.

Step	Procedure	Expected Result	Test Result
1	Change to "Measurements" tab	The program allows the user to change between tabs	Passed ✔
	Check if the data is being displayed in the waveform chart when is being taken	The waveform chart is updated with every sample taken	Passed ✔
	Check if the data is being displayed in the measurements table	The table is updated with every sample taken	Passed ✔
2	When the program is stopped, check if the report has been generated in the indicated path	The report must be generated with the name of the sample	Passed ✔

Table 26:UC 5 TEST

7.8.2 Non-Functional requirements test

7.8.2.1 Non-functional requirement 1: Motion

Description: The test checks if the response of the motors is fast enough.

Step	Procedure	Expected Result	Test Result
1	Check if the delay of the order to move the motors is less than 2s	The motors delay is less than 2s	Passed ✔

Table 27:MOTION TEST

7.8.2.2 Non-functional requirement 2: Voltage

Description: The test checks if the voltage value is true and relates the value in the correct representation.

Step	Procedure	Expected Result	Test Result
1	Check if the accuracy of the sampled values is sufficient (accuracy of 0.005 V)	The values are in the established range (accuracy of 2.19µV)	Passed ✔

Table 28:VOLTAGE TEST

7.8.2.3 Non-functional requirement 3: Temperature

Description: The test checks if the operational temperature of the drivers is acceptable.

Step	Procedure	Expected Result	Test Result
1	Check the Vref in the drivers to know if the driver is burned	The drivers must work correctly	Passed ✔ (was inserted a heatsink)

Table 29:TEMPERATURE TEST

8 RESULTS & DISCUSSION

At the section [**2. INTRODUCTION**] of this report, I have set forth the goals that I wanted to achieve. In this section the accomplished results are overviewed and discussed.

8.1 CREATION OF A LABVIEW APPLICATION (VIRTUAL INSTRUMENT (VI))

The LabVIEW application was designed to show the voltage of different test point of the pcb under test to the user, manage the data and make a report with them. As a result, it has been created a user-friendly layout that can be understandable for the common user. The application shows the voltage in real time and gives the possibility to change, as the case may be, the sample rate and sample time of acquisition values. Furthermore, it creates a report with all the data collected, the average, the maximums and the minimums of such data as well as an image of the test points tested to complement a rigorous report.

8.2 SYSTEM BUILDING

Initially, I want to build a System by myself but I have faced with the time limitation. Instead, I have bought the pieces in specialized store and just had to mount the structure and components.

8.3 ELECTRONICAL

The electronical part was the easiest one because of the components are really similar to the mounted in 3D printers or CNC machines. I made a wire connections between the components to make it easier connection to Arduino microcontroller board. However, I have chosen wrong the motors because I realized that they didn't had enough strength in order to move the Z axis but I have solved that problem including anther motor in such Z axis.

9 CONCLUSION

I have worked during a semester with the purpose of creating a smart system that will support the users saving time and avoiding the repetitive work. While creating the project, I was having ideas to make it useful in the real-world situations i.e. do not all the samples have to be similar or ways of improving the final report generated by AMSOP.

The goal of the project was to automatize a measurement system by means of an application. I have succeeded in creating a user-friendly interface together with the AMSOP hardware part.

The acceptance test shows and proves that the main parts of the AMSOP such as LabVIEW application, electronical components and all of them as a whole, work properly. Most of the test show positive results. However, I have found some problems with the motor Z axis because of such motor has not enough strength to be able to move it by itself. A possible solution is adding another motor working in parallel with this one thus minimizing the required strength in each motor.

Another possible problem that I found doing this project is that not everybody has the software and tools available to use the AMSOP, either by economic issues or because they do not need it. Because of that, I created another version that tried to simulate the machine's performance without the need to have the abovementioned expensive devices.

As a future improvement, as I said before, I have though to add another motor working at the same time in the Z axis. While I was doing the project, I have found that if you use LabVIEW, that is compatible with almost all the common lab instruments, you will be able to use AMSOP not only as a data collector but also as an instrument to program temperature cycles in a programmable climatic chamber or to stablish communication with a wideband radio communication tester. Due to all of these reasons, AMSOP is a very useful tool for test and validation areas of the companies.

I would finally appoint out that this project has been a personal challenge and the end of an import stage of my life, that has made me grow on a personal way and has formed me in order to face the future challenges that I will find in my professional career.

10 REFERENCES

1. “Arduino – LabVIEW communication” [Online]. Available: https://forums.ni.com/t5/Discusiones-sobre-Productos-NI/comunicar-labview-con-arduino. [Accessed: 11-Oct-2017].

2. “Different topics related with LabVIEW” [Online]. Available: https://forums.ni.com. [Accessed: from Oct to April].

3. “The SAD and its Architectural Views – Chapter 39, Documenting architecture: UML ¬ the N+1 View Model” [Online]. Available: https://bb.au.dk/bbcswebdav/pid-667268-dt-content-rid-1444382_1/courses/BB-Cou-UUVA61864/Documenting_Architecture_-_UML_and_the_N%2B1_View_Model.pdf [Accessed: 15-Feb-2018].

4. “DRV8825 comparaison” [Online]. Available: https://www.staticboards.es/blog/drv8825-vs-a4988. [Accessed: 12-Dic-2017].

5. “Chopping of drivers” [Online]. Available: https://www.luisllamas.es/motores-paso-paso-arduino-driver-a4988-drv8825. [Accessed: 20-Dic-2017].

6. “Stepper motors and LabVIEW” [Online]. Available: https://forums.ni.com/t5/LabVIEW/Arduino-Interface-for-LabVIEW-quot-Stepper-Write-VI-quot-Write/td-p/3757373 [Accessed: 13-Feb-2018].

7. “Project guide” [Online]. Available: https://blackboard.au.dk/webapps/portal/execute/tabs/tabAction?tab_tab_group_id=_130_1. [Accessed: 15-Oct-2017].

8. “System structure and components” [Online]. Available: http://www.teknlife.com/practico/como-montar-una-impresora-3d-parte-1-primeros-pasos. [Accessed: 04-Feb-2018].

9. “Bench power supply conversion” [Online]. Available: https://www.electronics-tutorials.ws/blog/convert-atx-psu-to-bench-supply.html. [Accessed: 1-Dic-2017].

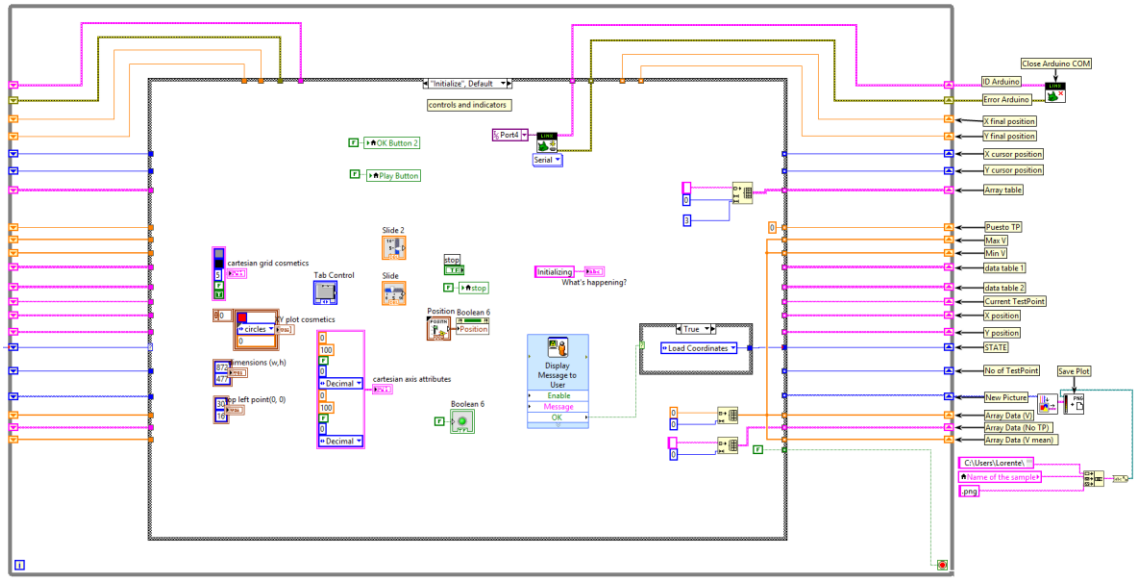
10. “Endstops switches” [Online]. Available: http://ezcontents.org/build-3d-printer-part-8-endstops. [Accessed: 04-Gen-2018].

11 ANNEXES

11.1 ANNEX I

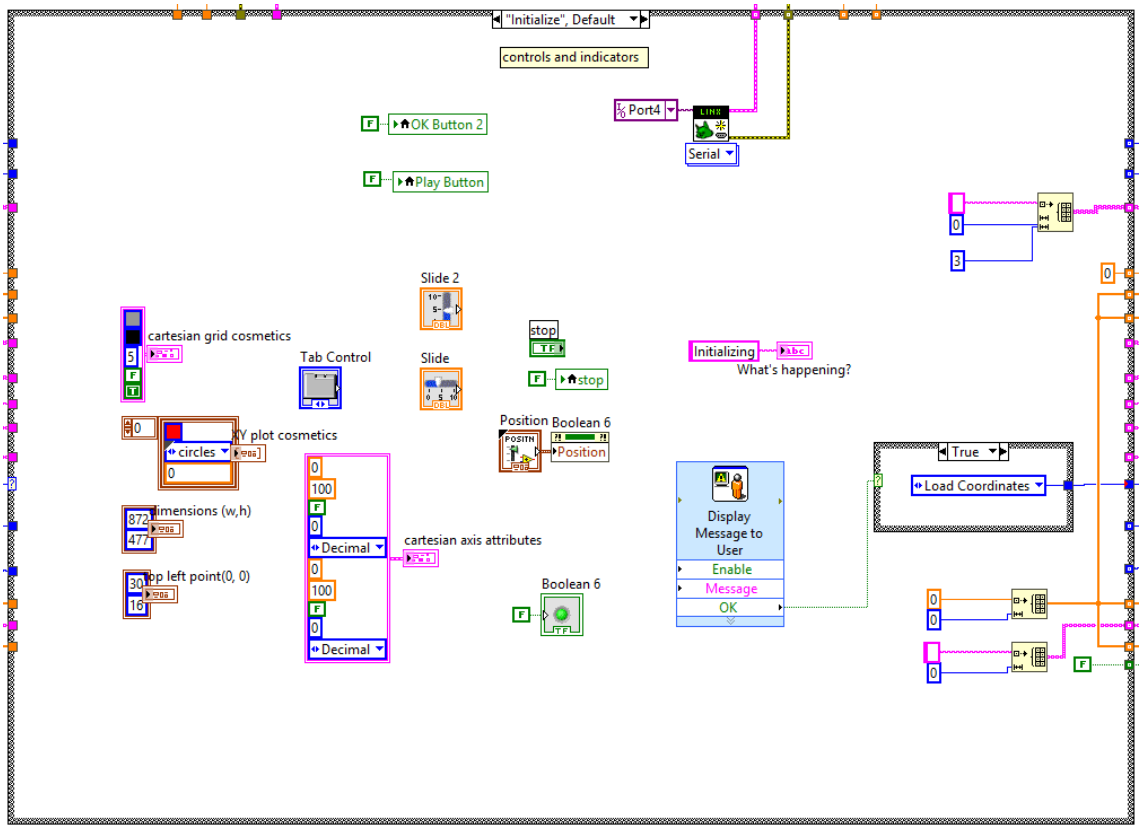
State machine code developed in LabVIEW

State machine block diagram:

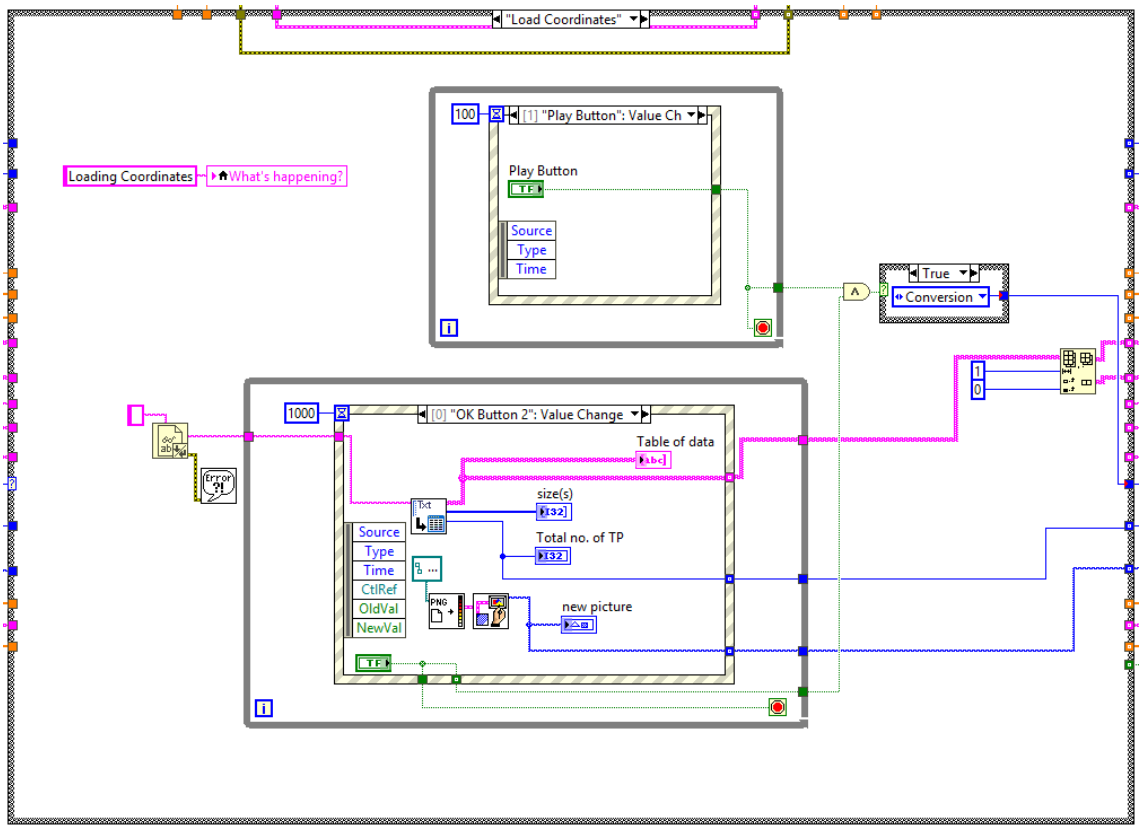


States:

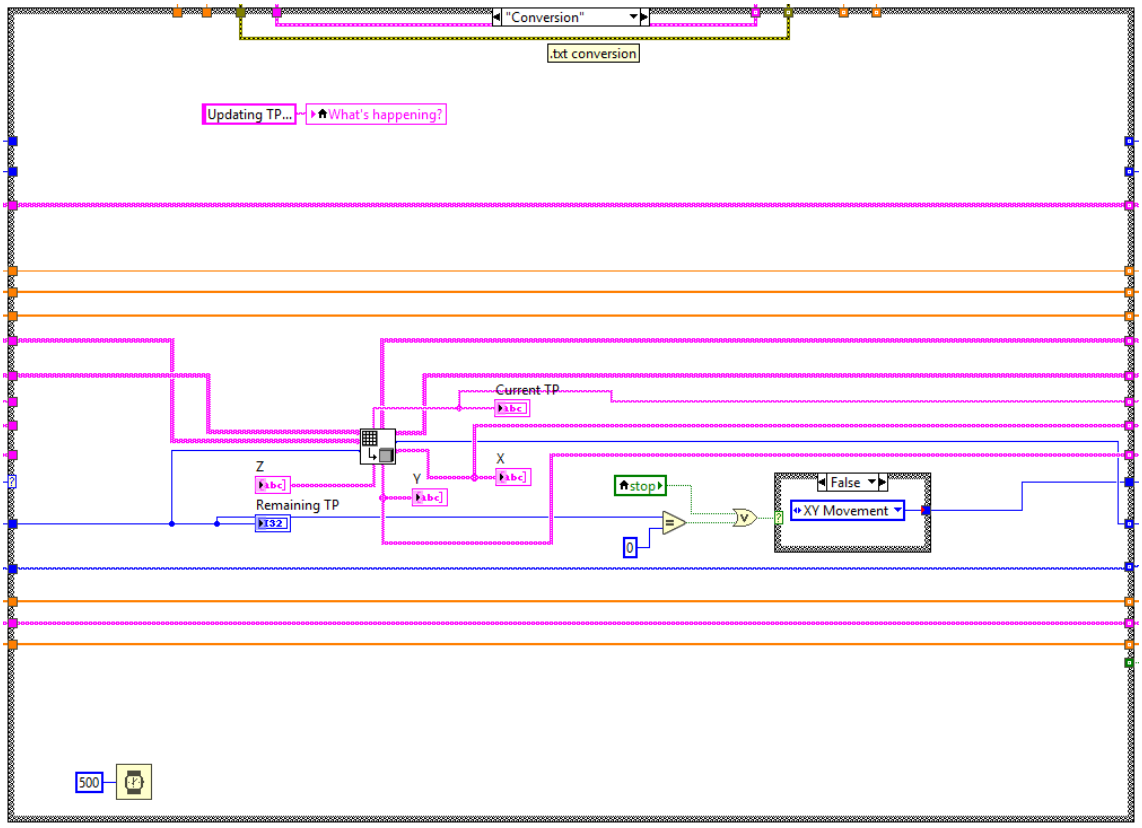
Initialize



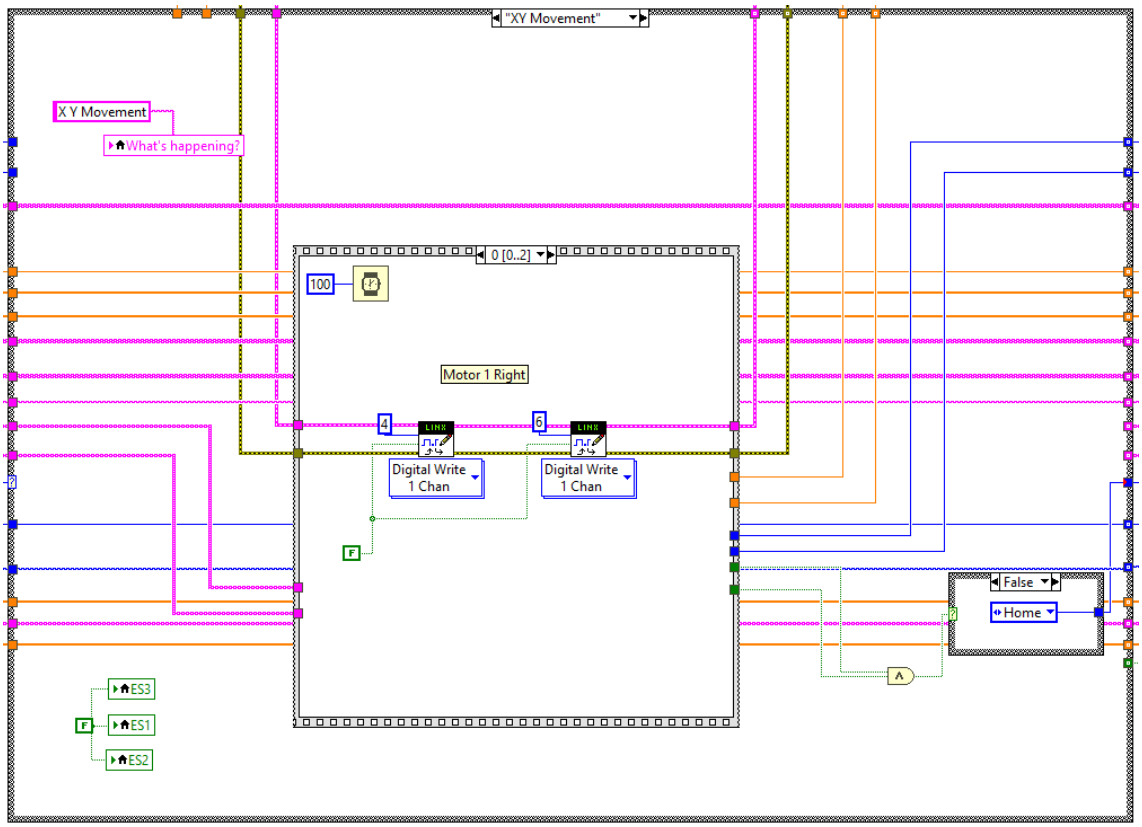
Load Coordinates



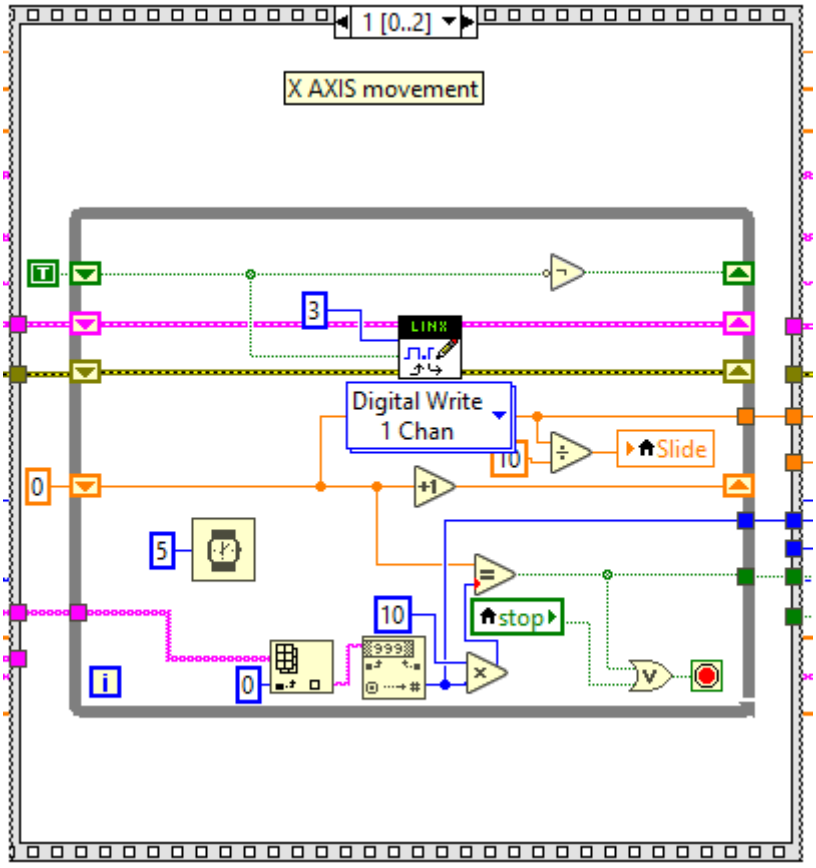
Conversion



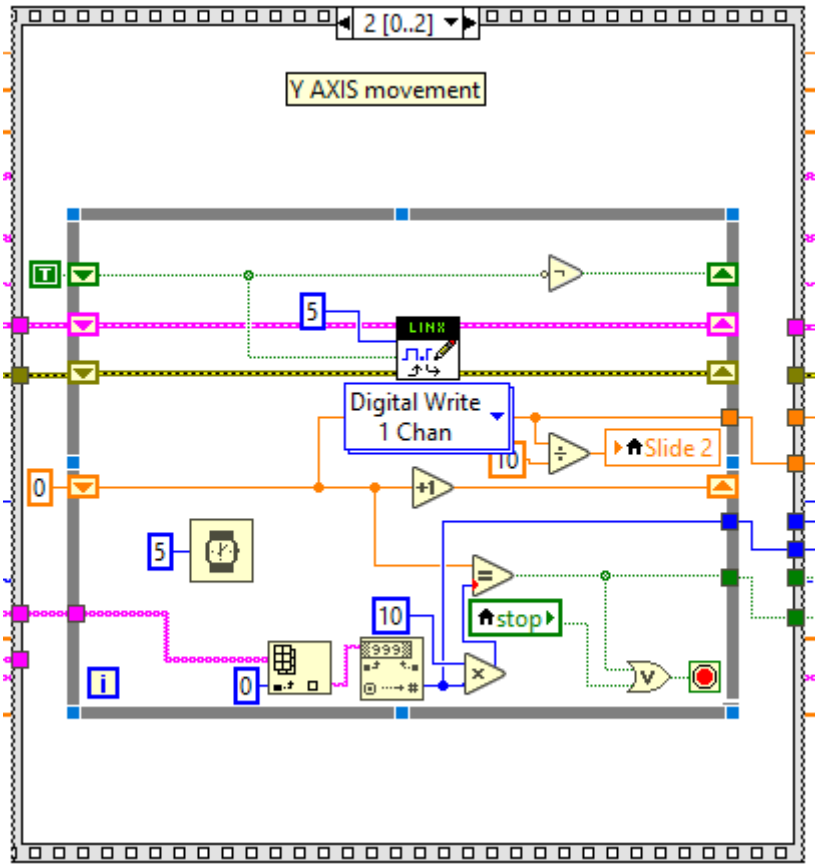
XY Movement



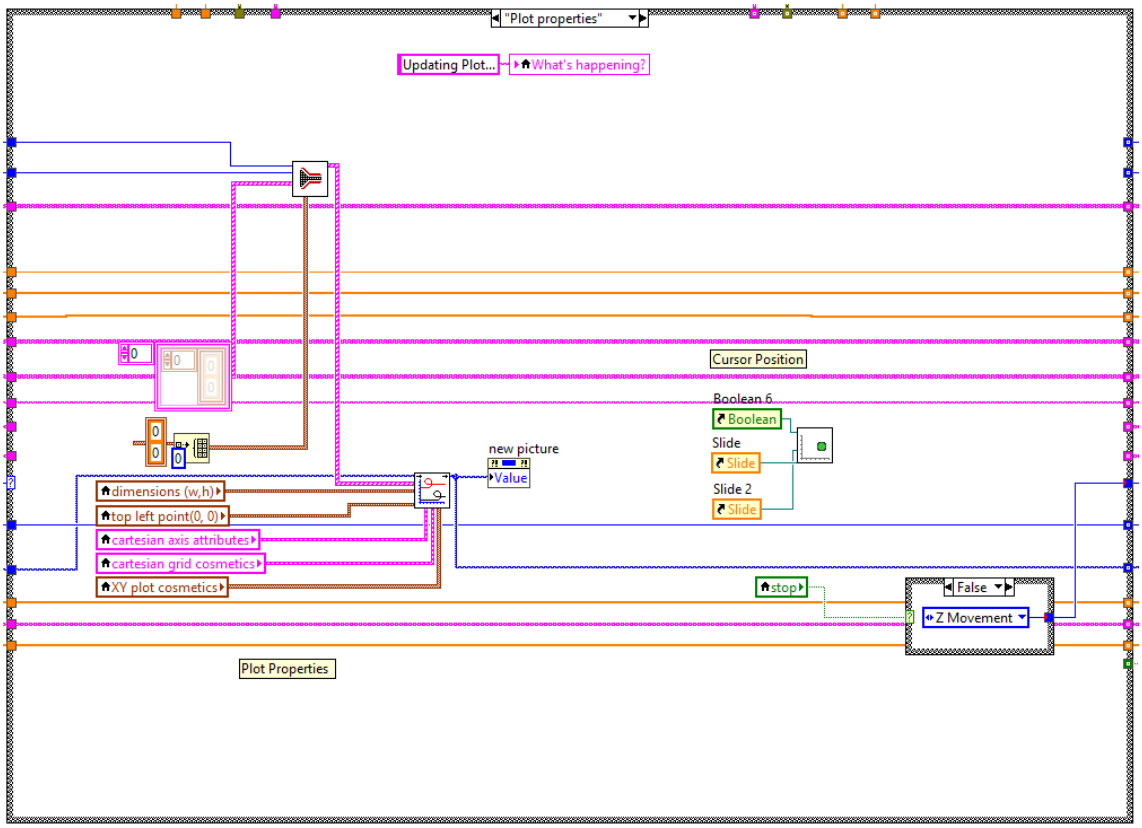
X Axis Movement



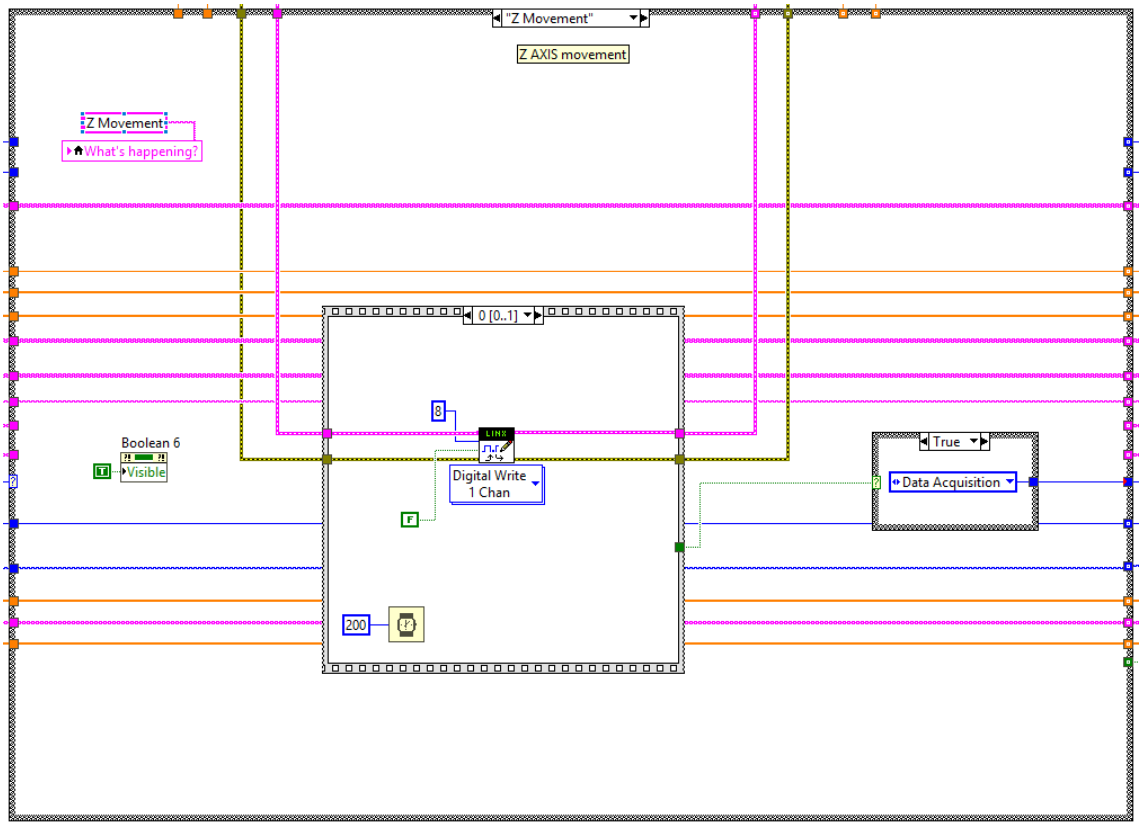
Y Axis Movement



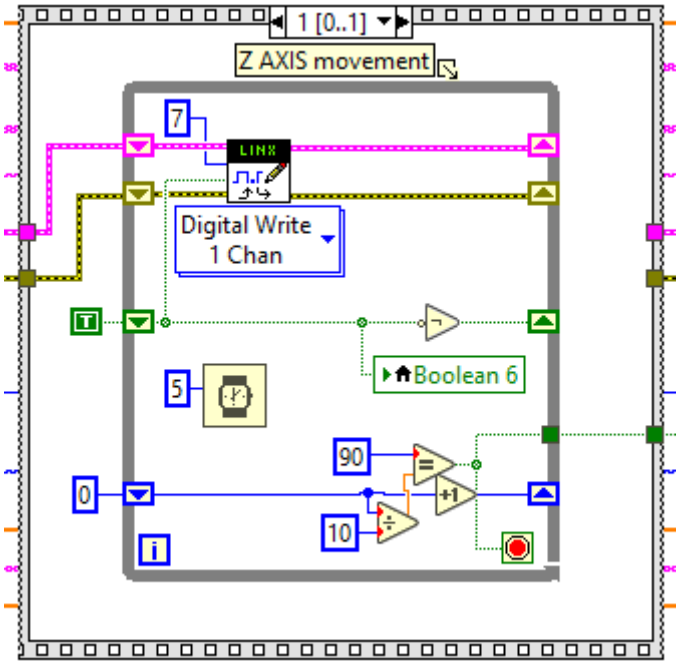
Plot Properties



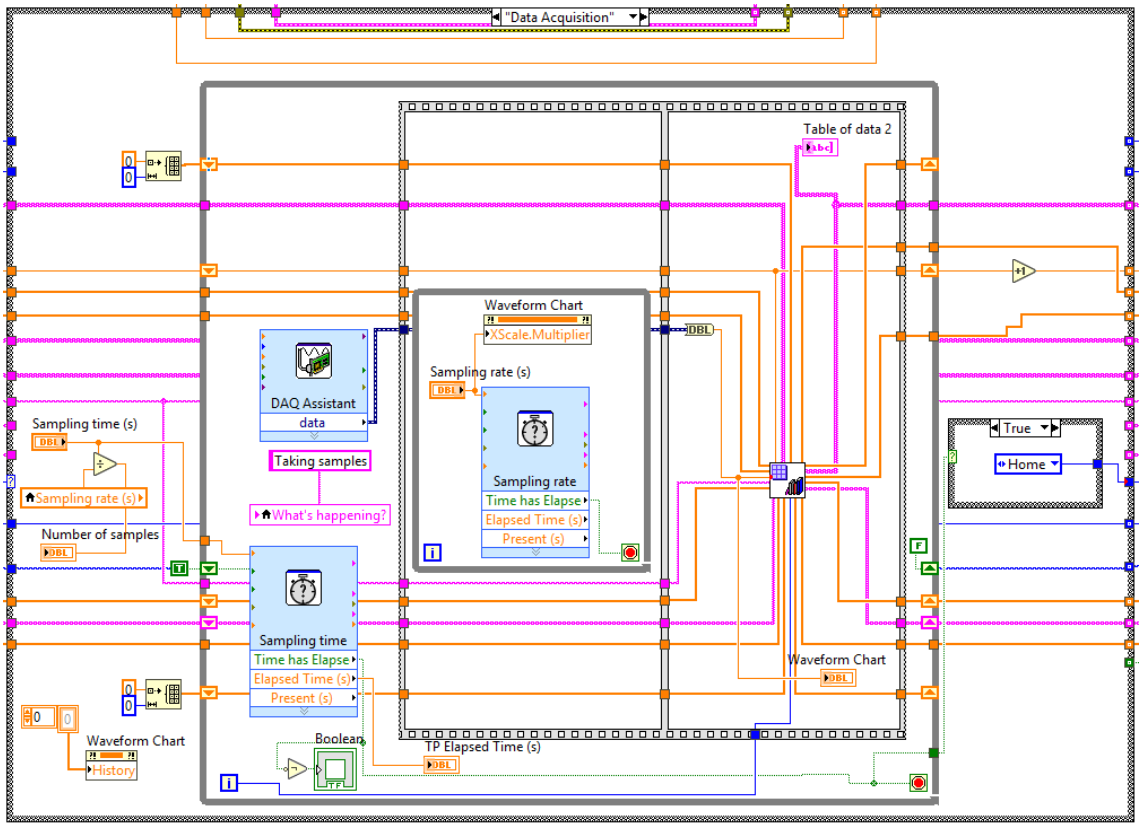
Z Movement



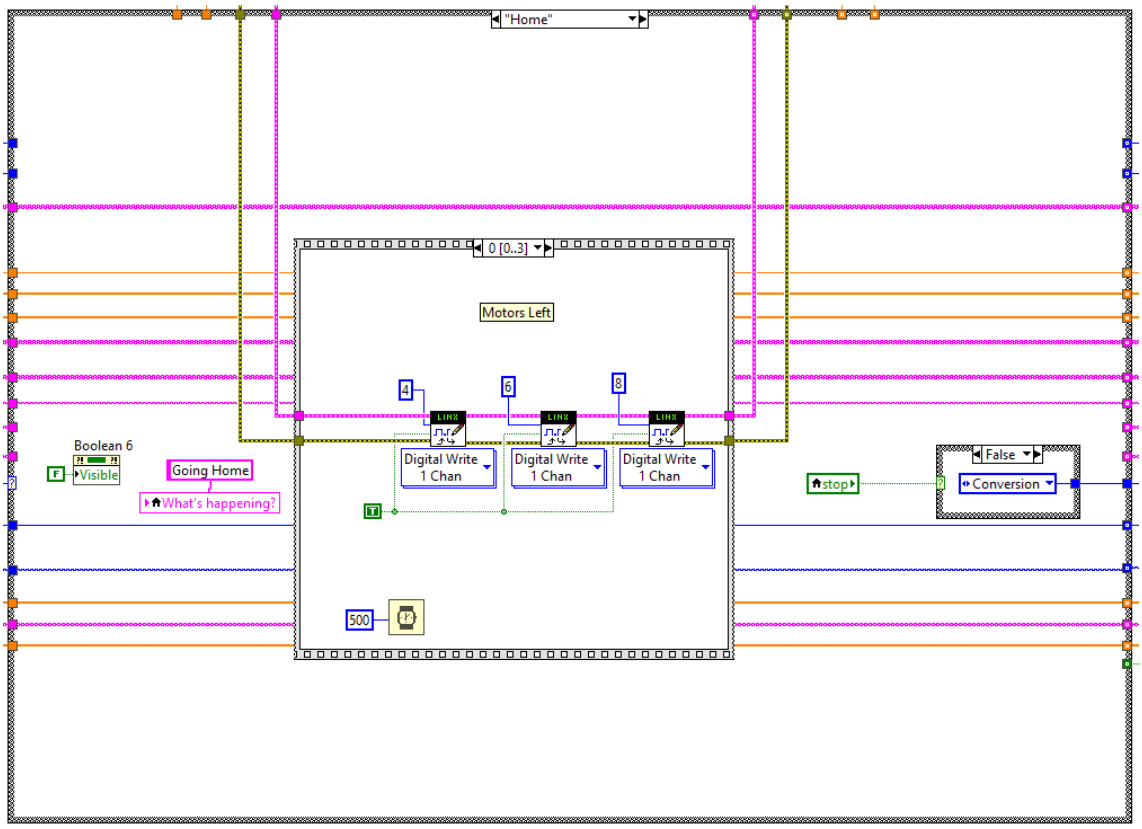
Z Axis Movement



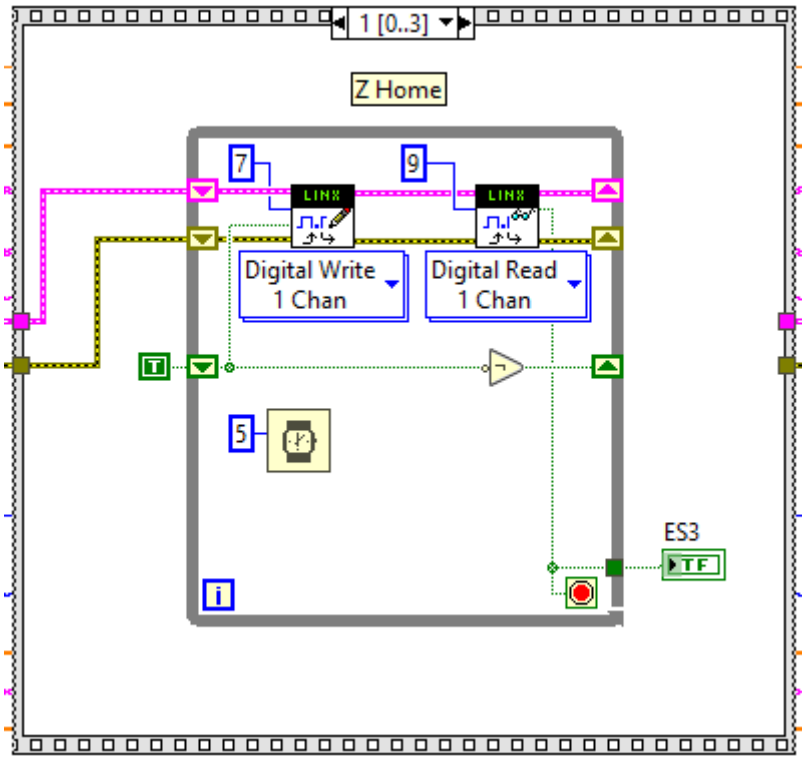
Data Acquisition



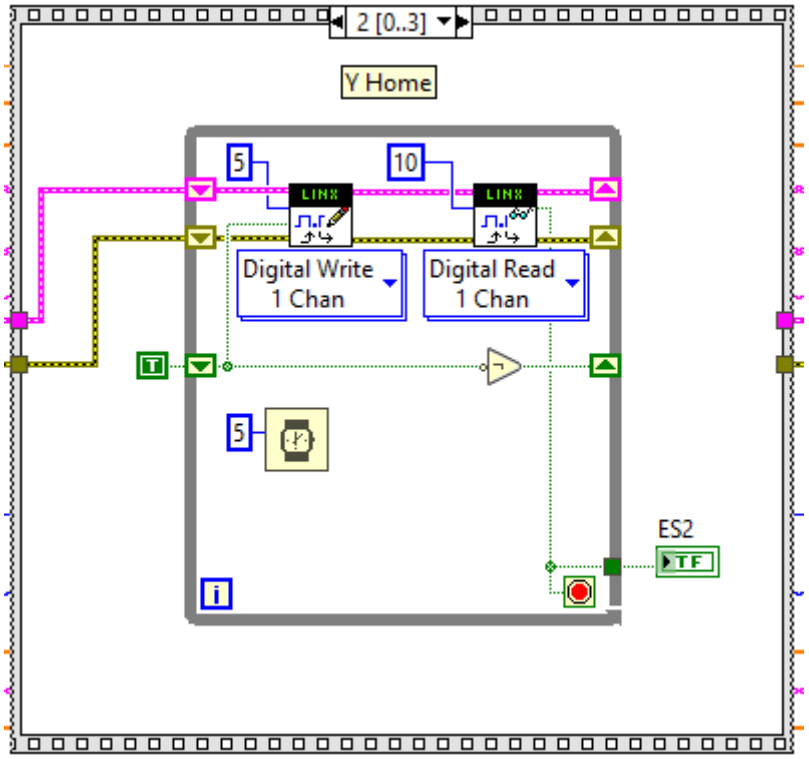
Home



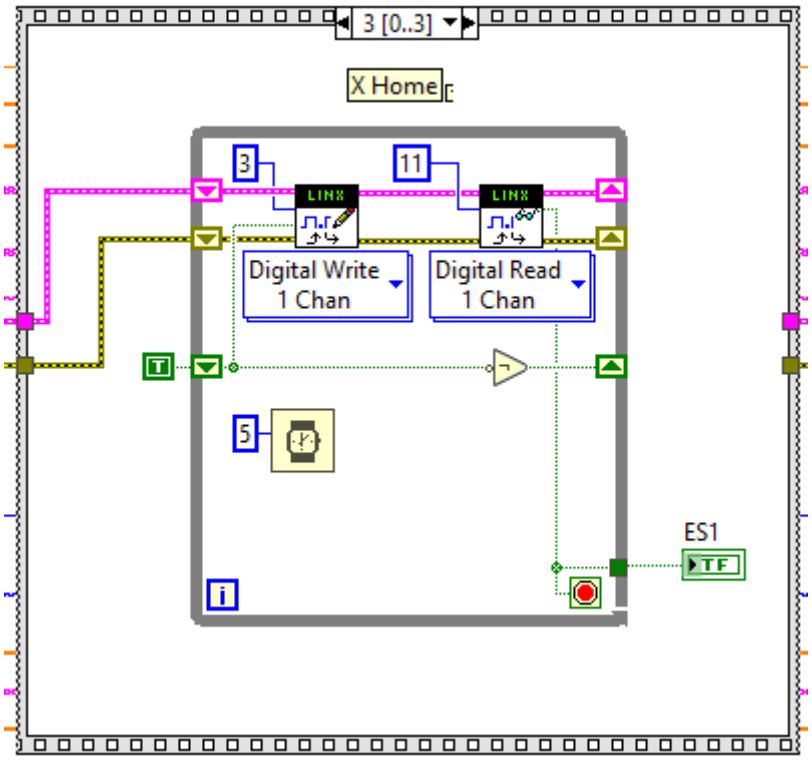
Z Home


1 [0..3]
Z Home
7
9
LINX
LINX
Digital Write
1 Chan
Digital Read
1 Chan
5
ES3
TF

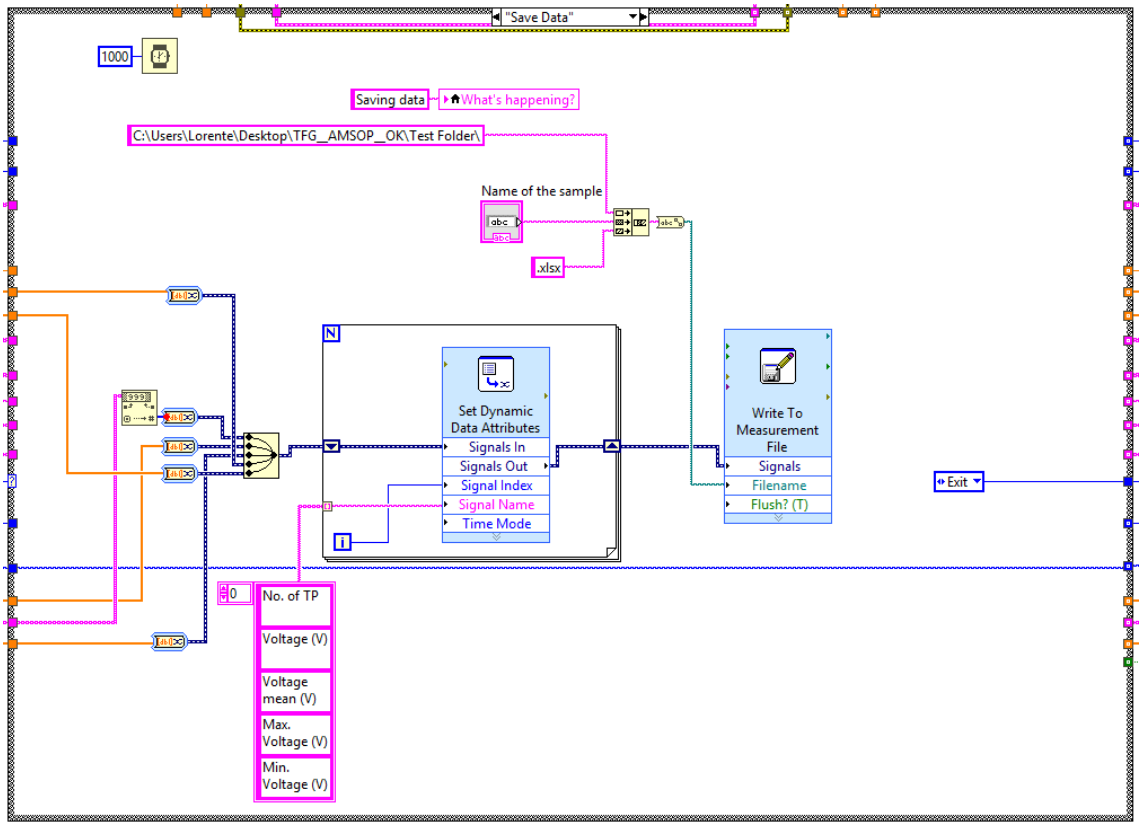
Y Home


2 [0..3]
Y Home
5
10
LINX
LINX
Digital Write
1 Chan
Digital Read
1 Chan
5
ES2
TF

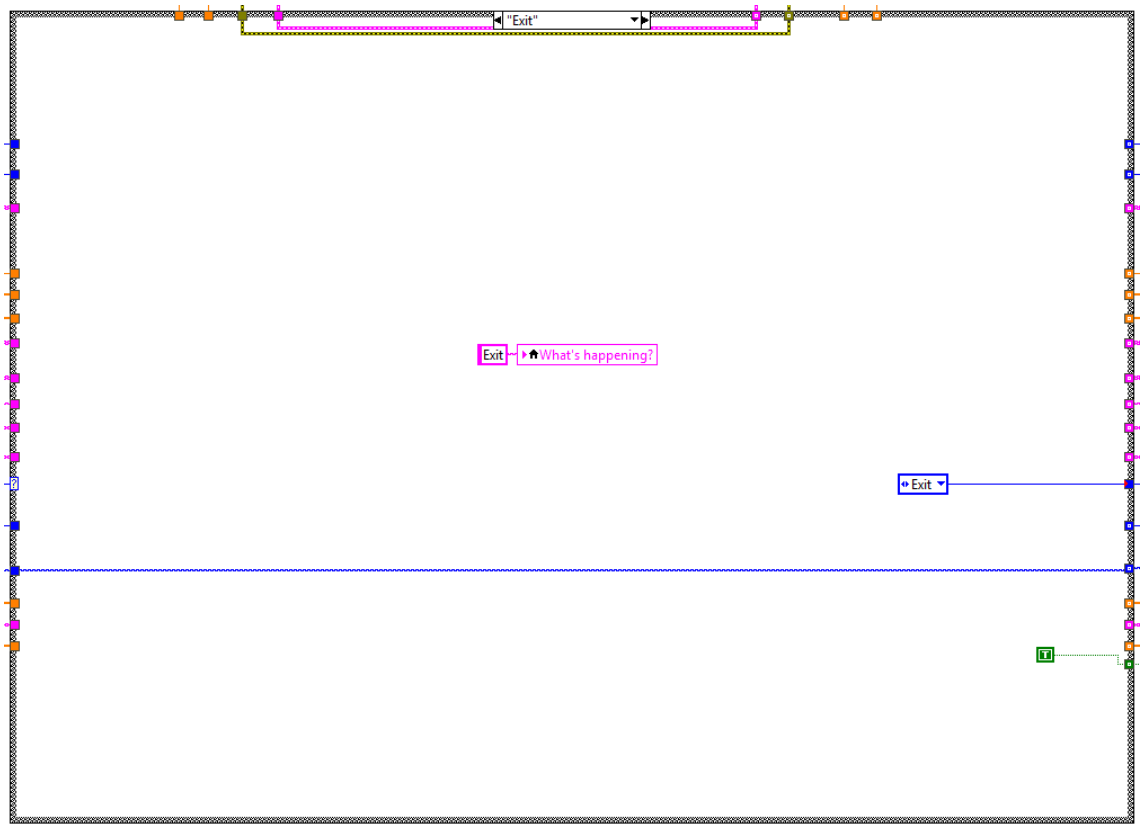
X Home



Save Data

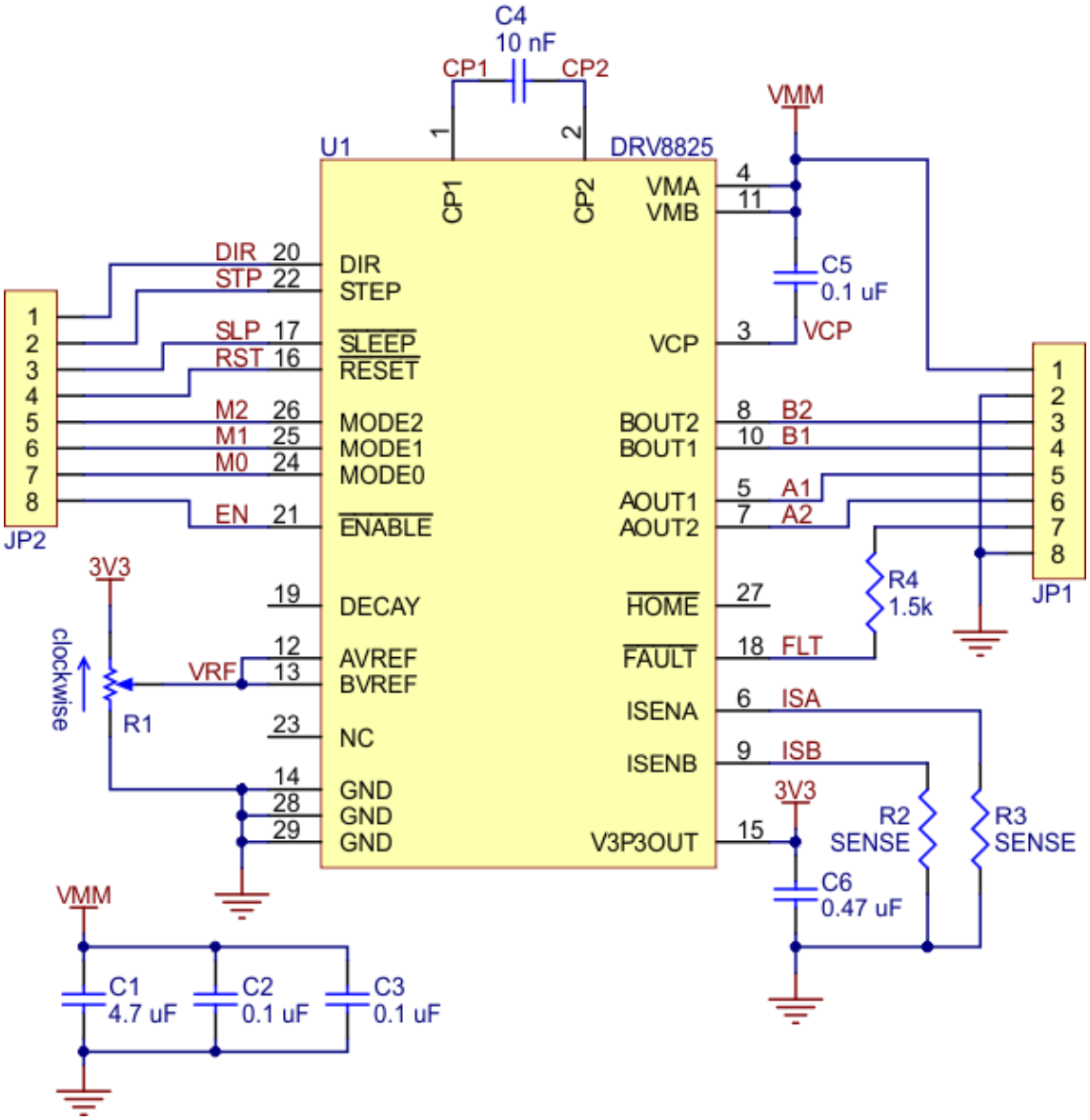


Exit



11.2 ANNEX II

Schematic diagram of the md 20ª DRV8825 stepper motor driver


C4
10 nF
CP1
CP2
U1
DRV8825
VMM
DIR
STP
SLP
RST
M2
M1
M0
EN
MODE2
MODE1
MODE0
SLEEP
RESET
ENABLE
DECAY
AVREF
BVREF
NC
GND
GND
GND
VMA
VMB
VCP
BOUT2
BOUT1
AOUT1
AOUT2
HOME
FAULT
ISENA
ISENB
V3P3OUT
C5
0.1 uF
VCP
B2
B1
A1
A2
R4
1.5k
FLT
ISA
ISB
3V3
R2
SENSE
R3
SENSE
C6
0.47 uF
JP2
JP1
3V3
clockwise
VRF
R1
C1
4.7 uF
C2
0.1 uF
C3
0.1 uF